CSIKÓS ZSOLT

ZENE KEDVELŐKNEK C 64/128

C 64/128
KEDVELŐKNEK
C 64/128
ZENE
ZENE

LSI ALKALMAZÁSTECHNIKAI
TANÁCSADÓ SZOLGÁLAT

ZENE C-64/128 KEDVELŐKNEK

írta: Csikós Zsolt
lektorálta: Németh István

A könyv rajzait és szedését Hajdú András készítette
Macintosh számítógéppel

LSI

LSI Alkalmazástechnikai Tanácsadó Szolgálat
Budapest, 1987

Kiadó: LSI ATSZ
Felelős kiadó: Dr. Kovács Magda
Témafelelős: Székely László
Engedélyszám: 51837
ISBN: 963 592 592 1
Készült az OMIKK házinyomdájában
Felelős vezető: Tóth Károly

Tartalomjegyzék

Előszó

Sokak szerint a számítógéppel nem lehet igazán zenélni, nem lehet a hangszereket tökéletesen utánozni. A zene számokkal való leírása embertelenné teszi a hangzását - mondják-, s ebben van is némi igazság.

A számítógéppel való zenélés nem pótolhatja az énekkarokat, szimfonikus zenekarokat, popegyütteseket. Célja nem is ez, hanem új, szokatlan hanghatások keresése, a zenei lehetőségek szélesítése. E kötet írásakor fontosnak tartottam, hogy mindenki számára érthető legyen, ezért csak a BASIC nyelv ismeretét tételeztem fel.* Remélem azonban, hogy a programokban előforduló trükkök a jártasabb olvasók számára is hasznosak lesznek. Arra törekedtem, hogy a könyv végére érve minden olvasó megbizonyosodjék: igenis lehet BASIC-ben is jó zeneprogramokat írni, ha az ember elég ügyes.

Természetesen magas szinvonalú zene- és beszélőprogramokat csak gépi kódban lehet írni. Ilyen például a "Sam/Reciter" című, mely bár elég gépies hangon, de beszél. (Védelmére legyen mondva, hogy az angol szavak kiejtése sokban különbözik írásuktól.)

A Ghostbusters program kis szellemének nevetése is igen hatásos, de az általam ismert programok közül az a legtökéletesebb, amely az "Everybody's kung fu fighting" c. dal kezdetét adja elő (énekli) olyan minőségben, mintha rádióból szólna. Szinte hihetetlen, hogy némi fáradsággal mi mindent ki lehet hozni a C-64-ből.

Természetes, hogy a jó játékprogramoknak is nélkülözhetetlen eleme a zene. Érdemes megemlíteni, hogy a popzene olyan nagy alakja is bekapcsolódott a software-gyártásba, mint Paul McCartney. A "Give My Regards to Broad Street" c. filmjének forgatókönyve alapján, az ő segítségével készült C-64-re egy játékprogram, melyben természetesen a film sikerdala is megszólal.

Bár kétségtelenül jó lenne már első próbálkozásra a fentiekhez hasonló programokat írni, de ehhez sajnos nem tudok receptet adni. Azok azonban, akik figyelmesen végigolvassák ezt a könyvet, és áttanulmányozzák a programokat, közelebb kerülnek a csúcsokhoz. (A kiadó tervezi, hogy ha e könyv sikeres lesz, megjelenteti a folytatást, amely a gépi kódú programozást, és a zenéléshez szükséges hardware kiegészítéseket mutatja majd be.) Sok olyan emberrel beszélgettem már, akik - bár jó programozók - zene programozásához hozzá sem mernek kezdeni, mert, mint mondják, nincs hallásuk. Számukra remélhetőleg nagy segítség lesz a kottaolvasás felelevenítése. Nem törekedhettem teljes részletességre, azonban az első fejezet elolvasása elegendő egy átlag kotta elolvasásához, számítógépbe táplálásához.

---

* Ha az olvasó úgy érzi, hogy még a BASIC nyelv terén is vannak hiányosságai, akkor ajánlatos e könyvvel párhuzamosan az irodalomjegyzék [1] -el jelölt művének első kötetét is tanulmányoznia. (Az [1] az irodalomjegyzékben szereplő első könyvet jelenti. A későbbiekben ezt a jelölést fogjuk használni.)

Kiknek ajánlom ezt a könyvet?

- Azoknak akik kevés fáradsággal bosszantani akarják szomszédaikat;
- azoknak, akik valamelyest ismerik a BASIC utasításokat, és szeretnék azokat ügyesen használni;
- azoknak, akik úgy érzik, mindent tudnak a programozásról, de erről bizonyosságot szeretnének nyerni;
- azoknak, akik arra kíváncsiak, hogyan lehet a Commodore 64-en zenélni, és hogyan lehet a zenét grafikával kísérni;
- végül mindazoknak, akik úgy érzik, nem tartoznak az előző csoportok egyikébe sem.

Végezetül szeretném megköszönni azt a segítséget, melyet a Marx Károly Közgazdaságtudományi Egyetem Számítástechnikai Intézetétől, a Chorus Oeconomicus énekkartól, lektoraimtól, barátaimtól kaptam.

Csikós Zsolt

FIGYELEM !

A Commodore 128 számítógép C-64-es üzemmódjában az itt közölt valamennyi program működik.

A Commodore 128 számítógép C-128-as üzemmódjában (BASIC V7.0) az összes olyan BASIC program működik, amelyik csak a SID chip regisztereit használja. Ebben az esetben a dallamokat a gép lassabban játssza le, mert az interpreter lassabb mint a C 64-é. A FAST: RUN parancsok kiadása után viszont gyorsabbak lesznek a dallamok, mert 2 MHz-es órajellel üzemel a gép. A zene ritmusát megadó értékek módosításával korrigálni lehet a gép sebességéből adódó eltéréseket.

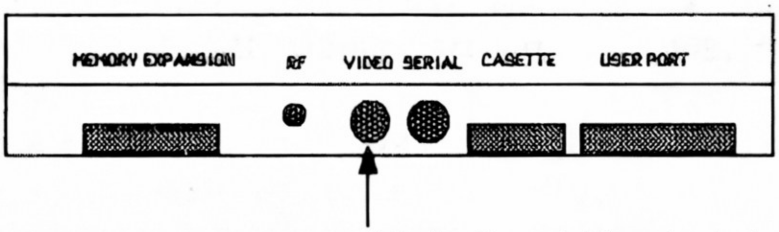
0. fejezet

E rövid fejezetben néhány lényeges tudnivalóra szeretném felhívni a figyelmet. Ezeket az előszó keretében is leírhattam volna - de meggyőződésem, hogy többen csak itt kezdik el olvasni a könyvet.

1./ Vannak olyan kisméretű televíziók, amelyek a C-64-gyel összekötve monitorként használhatók, zenélni mégsem tudunk velük.

2./ Bizonyos tv-készülékeken nem ugyanott a legtökéletesebb a hang és a kép. A zeneprogramok futtatásakor célszerű azt a sávot megkeresni, ahol a hang minősége a legjobb.

3./ Valószínű, hogy a programozók többsége nem lesz megelégedve a televízió hangjával. Ezért aki teheti, mindenképpen próbálja meg számítógépét erősítővel, vagy erősítőt tartalmazó lemezjátszóval, rádióval összekötni. (Videoton EA 6383 S erősítő esetén a TAPE 2 erősítőbemenetet használjuk.)



ide kell az átjátszózsinórt csatlakoztatni

Így akár hi-fi minőséget is elérhetünk. Mivel a tv-monitort és az erősítőt is egy helyre kell csatlakoztatni (ld. ábra), ezért zene és grafika együttes használata esetén alapkiépítésben le kell mondanunk az erősítőről. A csatlakozó átalakításával viszont (a technikai kérdésekre a későbbiekben még visszatérek) egyszerre is használhatjuk a monitort és az erősítőt.

Lehetőség nyílik sztereo jellegű hanghatások elérésére is. Egy többszólamú dal hallgatásakor a hangszínszabályozókat be lehet úgy állítani, hogy a tévén ill. az erősítőn más-más típusú hangok domináljanak. Ekkor a program futásakor úgy érezzük, mintha sztereo zene szólna. Ezt kipróbálhatjuk a 3. fejezetben ismertetésre kerülő "Tourdion", ill. az "Ébred már a..." c. dalokat játszó programokkal.

4./ Ne ijedjünk meg egy hosszú program begépelésétől addig, amíg alaposabban meg nem néztük a listáját, hiszen a legtöbb programban sok az egyező DATA-sor. Ilyenkor elegendő egy ilyen sort begépelni, s utána már csak a sorszámot változtatva, a RETURN billentyű megnyomásával bevinni a többit. Ha figyelmesek vagyunk, akkor rengeteg fáradságtól kímélhetjük meg magunkat, és nem megy el a kedvünk az egész programozástól. Időt nyerhetünk az utasítások rövid formáját használva is, pl. a PRINT helyett ?-et, a POKE helyett PΓ-t, stb.

5./ Hosszabb programok írásakor inkább többször mentsük ki munkánkat készítés közben, minthogy rengeteg fáradságunk veszendőbe menjen egy kellemetlen véletlen miatt. Ne feledjük, az áramszünet mindig akkor jön, amikor már lényegében kész a program, csak még nincs kimentve!

Az is előfordulhat, hogy megírunk egy hosszabb programot, amelyben pl. a következő sor szerepel:

```
FOR I = 0 TO 24 : POKE S + I, 0 : NEXT
```

Ha véletlenül az S értéke már nem 54272 (a SID kezdőcíme), hanem valami más, akkor futtatáskor előfordulhat, hogy a program elakad, a számítógép kezelhetetlenné válik, s a kikapcsolásával elvész, amit beírtunk. Nem érhet viszont ilyen kellemetlenség bennünket, ha a futtatás előtt kimentjük programunkat.

Zeneelméleti alapismeretek és érdekességek

1.1 Hangrendszerek

Volt idő, amikor az emberek még nem ismerték a beszédet. A beszéd kialakulása után pedig sok-sok évnek kellett eltelnie ahhoz, hogy a kimondott szavakat írásban is rögzíteni tudják.

Az írás alapja legtöbb esetben egy betűkészlet: az ábécé. Ebben hasonlít egymásra az írott szöveg és a zenei kotta. Hiszen ahhoz, hogy egy-egy dallamot le lehessen jegyezni, szintén szükség van valamilyen ábécére - a hangok jelrendszerére.

Az ókori görögök jelölték először az egyes hangokat, azok magasságát az írott ábécé betűivel. Az általuk használt legmélyebb hangnak adták az "a" nevet, a következő lett a "b", és így tovább, végül a hetedik hangot a "g" jelölte. Persze nemcsak hét hangot használtak. Azt tapasztalták azonban, hogy a következő hét hang sorozata az előzőeket ismétli, egy kicsit magasabban - ezért a nyolcadik hang ismét "a" lett, a kilencedik "b", stb. (Ezekre azt mondjuk, hogy egy oktávval magasabbak az előző 7 hangnál.)

Az idő múlásával a "c"-vel kezdődő hangsorok használata vált uralkodóvá, így a zenei ábécé, az alapskála is módosult: c,d,e,f,g,a,b. Az utolsó hang magassága ekkor még nem volt pontosan meghatározva, "b"-vel kétféle hangot is jelöltek. Ezért később az egyértelműség kedvéért a skála utolsó hangját sok országban - Magyarországon is - átkeresztelték "h"-ra, "b"-nek pedig az "a" és "h" közti hangot nevezték. Így a zenei ábécé mai formája a következő:

c d e f g a h

Ezeket nevezzük törzshangoknak. Sok zeneműben viszont olyan hangok megszólaltatására is szükség volt, melyeknek magassága nem egyezett meg egyik skálahang magasságával sem. Ezért volt szükség a "b" hang bevezetésére is. A későbbiekben a skála hangjainál egy kicsivel alacsonyabb, ill. magasabb hangok jelölésére egységes elnevezés-rendszert vezettek be: az alacsonyabb hangot a törzshang nevéhez illesztett -esz, a magasabbat -isz jelzi.

Csökkentve	**Skálahang**	Felemelve
cesz	c	cisz
desz	d	disz
esz	e	eisz
fesz	f	fisz
gesz	g	gisz
asz	a	aisz
b[x]	h	hisz

[x] A "b" elnevezést hagyományból megtartották.

Az alfejezet további részében az tekintjük át, hogyan változott az egyes zenei hangokhoz tartozó hangmagasság értelmezése az idők folyamán. Szó lesz a Püthagorasz-féle kvintkörről, a Dimidosz által megalkotott diatonikus skáláról, valamint a napjainkban használatos egyenletesen temperált kromatikus skáláról. Mielőtt azonban elkezdenénk ezek elemzését, ismerkedjünk meg a konszonancia fogalmával.

Ahhoz, hogy egy hangot érzékeljünk, szükség van egy hangforrásra (egy alkalmas rezgő rendszerre, mint pl. egy húr), és valamilyen rugalmas közegre – ez többnyire a levegő – amely közvetíti a rezgést. A hang másodpercenkénti rezgésszámát frekvenciának nevezzük.

Ha két vagy több hang együttes megszólalása kellemes érzést kelt a hallgatóban, akkor konszonanciáról, ellenkező esetben - ha fülsértőnek halljuk - disszonanciáról beszélünk. A konszonancia csak a megszólaló hangok alapfrekvenciáitól, ezek egymáshoz való viszonyától függ. A konszonancia vizsgálatával kapcsolatban Püthagorasz nevét kell elsőként megemlíteni, aki nagy figyelmet szentelt a zene matematikai elemzésének. Kortársai szerint egy kovács munkáját figyelve vette észre, hogy a különböző kalapácsok hangjai konszonánsak egymással. Megmérte a kalapácsok súlyait, s úgy találta, hogy ezek aránya 4:3, 3:2, ill. 2:1. Később szerkesztett egy egyhúros hangszert. Ez a hangszer - a monochord - lényegében egy 12 egység hosszú, kifeszített húr. Amikor úgy fogta le a húrt, hogy annak rezgő része 6, 8, ill. 9 egységnyi volt, akkor szintén azt tapasztalta, hogy a megszólaló hangok konszonánsak egymással. A Püthagorasz által talált konszonáns hangpárok frekvenciáinak aránya a következők (a megszólaló hang frekvenciája fordítottan arányos a rezgő húrrész hosszával):

$$\begin{aligned} 12:6 &= 2:1 \quad (\textit{oktáv}) \\ 12:8 &= 3:2 \quad (\textit{kvint}) \\ 9:6 &= 3:2 \\ 12:9 &= 4:3 \quad (\textit{kvart}) \\ 8:6 &= 4:3 \end{aligned}$$

Püthagorasz azt állította, hogy a hangsorok[*] elemei előállíthatók pusztán a kvint segítségével (kvintkör). Módszerének lényege a következő: egy alaphangból kiindulva kétszer kvint fölfelé, egyszer oktáv lefelé, s ezt ismételjük megfelelően sokszor. Egy ilyen kör után a kapott hang frekvenciája a következőképpen számítható:

$$f_1 = f_0 \cdot \frac{3}{2} \cdot \frac{3}{2} \cdot \frac{1}{2} = f_0 \cdot \frac{9}{8}$$

kvintek felfelé (az első két tényező) – oktáv lefelé (a harmadik tényező)

(f_0 a kiindulási hang frekvenciája)

[*] Hangsornak az olyan hangok sorozatát nevezzük, melyeknek egymás utáni, vagy "egyszerre" történő megszólaltatása kellemes érzetet kelt.

A módszer, bár első látásra jónak tűnik, mégsem tökéletes. A ciklus hatodik ismétlése után a kezdeti hangnál éppen egy oktávval magasabbra kellene eljutnunk, vagyis $f_6 = f_0 \cdot 2$ kellene, hogy fennálljon. Ehelyett azonban az eredmény

$$f_6 = f_0 \cdot \left(\frac{9}{8}\right)^6 \approx f_0 \cdot 2{,}0273 \quad \text{lesz.}$$

Ezt a problémát küszöbölte ki Dimidosz, aki az i.e. I. században élt. Az általa megalkotott diatonikus (dúr) skála a következő:

	prím	szekund	nagy terc	kvart	kvint	nagy szext	nagy szeptim	oktáv
a hang neve	c	d	e	f	g	a	h	c
a hang és a prímhang frekvenciájának aránya	1	$\frac{9}{8}$	$\frac{5}{4}$	$\frac{4}{3}$	$\frac{3}{2}$	$\frac{5}{3}$	$\frac{15}{8}$	2
két szomszédos hang frekvenciájának aránya	$\frac{9}{8}$	$\frac{10}{9}$	$\frac{16}{15}$	$\frac{9}{8}$	$\frac{10}{9}$	$\frac{9}{8}$	$\frac{16}{15}$	

Lényeges szempontja egy zenemű előadásának az is, hogy egy adott dallamot ne csak egyetlen hangról lehessen elkezdeni, hanem bármelyikről. A diatonikus skálát használva ezt nem tehetjük meg, hiszen az egymást követő egész hangok aránya hol 10/9, hol 9/8, hol pedig 16/15. Emiatt az egymáshoz közeli hangok frekvenciáit egységesítették. Így hangsorunk a következő hangokból áll oktávonként: c,d,e,f,g,a,h, az ezekhez képest 25/24-szeres frekvenciájú cisz, disz, eisz, fisz, gisz, aisz, ill. a 24/25-szörös frekvenciájú cesz, desz esz, fesz, gesz, asz b. Mivel ez a 21 különböző hang bizonyos hangszereken - pl. zongora - nem szólaltatható meg, ezért kitalálták az ún. egyenletesen temperált kromatikus hangsort (Werckmeister, 1691). Az oktávnyi hangterjedelmet 12 részre osztották, úgy, hogy bármely két szomszédos hang frekvenciájának aránya
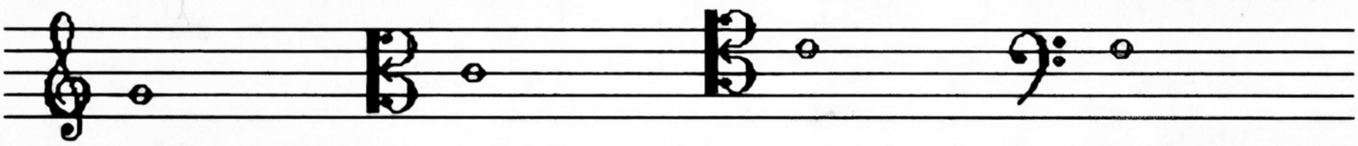
$$2^{\frac{1}{12}} = \sqrt[12]{2} \quad \text{legyen.}$$

Így a félhangok közötti különbségek eltűnnek, és a következő 12 alaphanghoz jutunk:

	cisz		disz	e	f	fisz		gisz		aisz	h	c_1
c		d					g		a			
	desz		esz	fesz	eisz	gesz		asz		b	cesz	hisz

a) Kottavonalak, kulcsok

Az újkori európai zene 97 különböző magasságú hangot, azaz lényegében nyolc oktávot használ. Az ötvonalas kottasoron csak 11 különböző hang jelölhető, ezért szükségessé válhat pótvonalak használata is. Persze a 97 különböző hang ábrázolásához igen sok pótvonalra lenne szükség. Ennek elkerülésére találták ki a különböző kulcsok használatát. A kottákban leggyakrabban violinkulcsot (G-kulcsot) és basszuskulcsot (F-kulcsot) használnak, de találkozhatunk kétféle C-kulccsal is (altkulcs és tenorkulcs). A kulcsok elnevezései és írásmódjuk összhangban van egymással, mindegyik kulcs a nevének megfelelő hang helyét határozza meg a kottában. (valójában nemcsak ez a két C-kulcs létezik, de a C hang helye mindig hasonlóan határozható meg.)



g^1 c^1 c^1 f

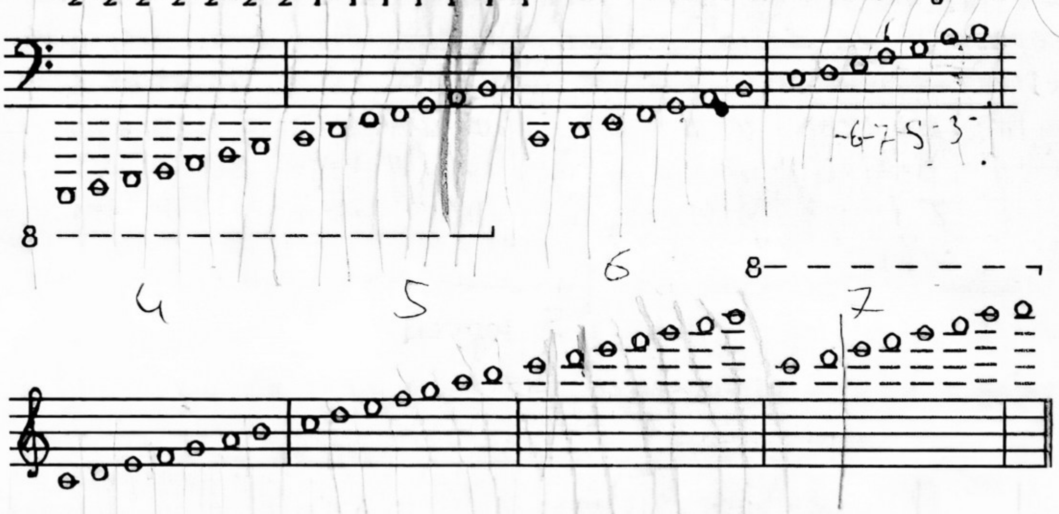
violinkulcs altkulcs tenorkulcs basszuskulcs

2. ábra

Amennyiben a megfelelő kulcs ellenére is sok pótvonalra lenne szükség - mint pl. a két szélső oktáv esetében -, akkor e hangok fölé vagy alá egy 8 ¬ ill. 8 ⌐ jel kerül, azt mutatva, hogy ezeket a hangokat a kotta szerintinél egy oktávval magasabban, ill. alacsonyabban kell játszani. A példa kedvéért ábrázoltuk a 8 oktávot.

oktáv: szubkontra kontra nagy kis

$C_2 D_2 E_2 F_2 G_2 A_2 H_2$ $C_1 D_1 E_1 F_1 G_1 A_1 H_1$ C D E F G A H c d e f g a h



3. ábra

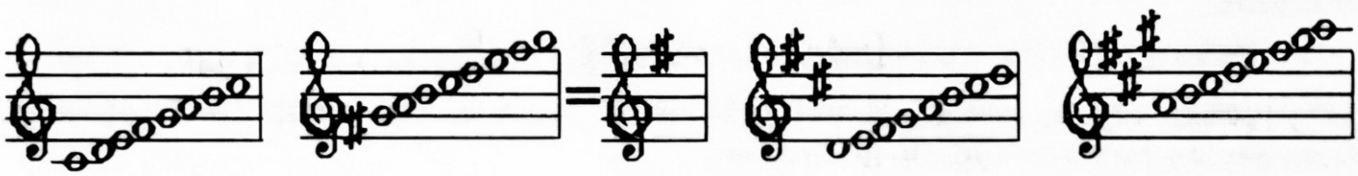
$c^1 d^1 e^1 f^1 g^1 a^1 h^1$ $c^2 d^2 e^2 f^2 g^2 a^2 h^2$ $c^3 d^3 e^3 f^3 g^3 a^3 h^3$ $c^4 d^4 e^4 f^4 g^4 a^4 h^4$ c^5

egyvonásos kétvonásos háromvonásos négyvonásos ötvonásos

b) Előjegyzések

A következő kérdés, hogy hogyan jelöljük a félhangokat. Ezt a célt szolgálják az előjegyzések. Mint korábban volt már róla szó, a törzshangokat (c,d,e,f,g,a,h) fölfelé és lefelé is módosíthatjuk. (Kromatikus hangsor esetén a kétféle módosítás mindig helyettesíthető egymással.) A kottákban is kétféle módosító jellel találkozhatunk: a kereszttel (♯), és a bével (♭). A kereszt egy fél hanggal emeli, a bé pedig egy fél hanggal csökkenti a módosítani kívánt hangmagasságot. (Létezik még kettős kereszt /×/ és kettős bé /♭♭/ melyek két fél hanggal módosítanak.) A módosítójelek felesleges ismétlésének elkerülésére a zeneszerzők, kottaírók sokszor közvetlenül a kulcs után teszik ki az előjegyzéseket, s így azok az összes megfelelő hangra vonatkoznak.

C-dúr G-dúr D-dúr A-dúr



4. ábra

G-dúr: f → fisz

D-dúr: f → fisz, c → cisz

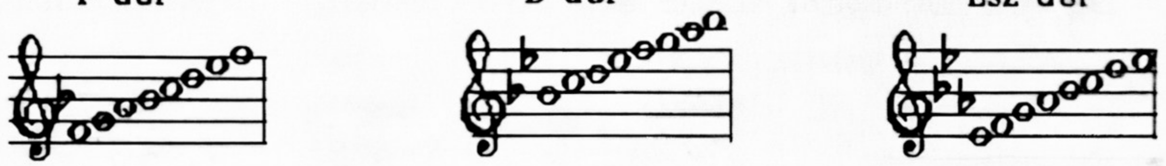
A-dúr: f → fisz, c → cisz, g → gisz

Például egy kereszt előjegyzés esetén azért nevezzük a hangsort G-dúrnak, mert az itt ábrázolható hangok egy olyan dúr hangsort alkotnak, melyet a G hangot dó-nak választva tudunk leszolmizálni.

G		A		H		C		D		E		FISZ		G
	1		1		1/2		1		1		1		1/2	
dó		ré		mi		fá		szó		lá		ti		dó

Az újabb és újabb keresztek behúzása nem esetleges, hanem jól meghatározott. Ahhoz, hogy egy hang felemelésével ismét dúr hangsort kapjunk, a jelenlegi fá hangot kell felemelni. Így pl. C-dúr esetében (nincs előjegyzés) az F hang a fá, ezért ezt kell a kereszttel felemelnünk, hogy megkapjuk a G-dúrt. A már említett különbséget leszámítva, hasonló a helyzet a bével kapcsolatban is.

F-dúr B-dúr Esz-dúr



F-dúr: h → b

B-dúr: h → b, e → esz

Esz-dúr: h → b, e → esz, a → asz

5. ábra

a./ 4 ill. 5 kereszt előjegyzés esetén hová kell tenni a 4. ill. 5. keresztet? Hol lesz a dó?
b./ Keressünk szabályt a b-vel történő módosításokra, majd ennek alapján növeljük az előjegyzésben szereplő b-k számát!

Bizonyos esetekben szükség lehet egyes hangok módosítására a kottán belül is. Az így kitett módosítójelek csak egy ütemre érvényesek. Előfordulhat az is, hogy semlegesíteni szeretnénk egy, a kotta elején kitett módosítójelet. Erre szolgál a feloldójel /♮/. Fontos tudni, hogy ez is csak egy ütemre vonatkozik. Kórusművekben találkozhatunk zárójelbe tett feloldójellel is /(♮)/, ami furcsának tűnik, mert ilyenkor nincs is mit feloldani. Ez a jel csak az énekkarokat figyelmezteti: ha egy akkordban az egyik szólamnak félhang módosítása volt, akkor a következő ütemben a többi szólam nehogy félhangot csússzon.

A dúr hangsorok mellett a másik uralkodó hangsor a moll hangsor. Ennek több fajtája létezik. Pl. a természetes moll hangsorban az egymásutáni hangok távolságai a következők:

$$1-1/2-1-1-1/2-1-1$$

Ez lényegében olyan, mintha lá-val kezdenénk szolmizálni, következésképpen pl. 1 kereszt esetén beszélhetünk e-mollról is.

Természetesen létezik több más hangsor is, pl. a magyar népdalok zömére a d-f-g-a-c ötfokú (pentaton) hangsor a jellemző.

c) *Hangjegyértékek*

Most már tudjuk, hogy egy-egy hangjegy milyen hangot, hangmagasságot jelöl, de azt még nem, hogy ezeknek mennyi ideig kell szólniuk. Csökkenő sorrendben a hangjegyértékek a következők:

	egész	fél	negyed	nyolcad	tizen-hatod	harminc-ketted
hang	𝅝	𝅗𝅥	♩	♪	𝅘𝅥𝅯	𝅘𝅥𝅰
szünet	𝄻	𝄼	𝄽	𝄾	𝄿	𝅀

6. ábra

5 4 3 2 1 0

(A szünet jelek azt mutatják meg, hogy az egyes hangszereknek, ill. szólamoknak mennyi ideig kell hallgatniuk a következő hang megszólalásáig. Ha több azonos értékű

* A továbbiakban Mfeladattal jelöljük azokat a feladatokat, melyeknek megoldása megtalálható a 2. függelékben.

hangjegy követi egymást, akkor az egyszerűség kedvéért gerendával kötjük össze őket. Pl.: 4 harminckettedes hang egymásutánja a következőképpen ábrázolható:



Különböző értékű hangjegyeket is összeköthetünk, pl.:

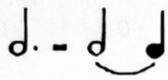


Természetesen egyéb hangértékek is elképzelhetők. Ezek leírására három jelölést használunk:
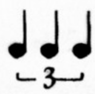
- kötőívvel összekapcsolt hangjegyek esetén az összekapcsolt hangjegyek értékei összeadódnak. Pl.:
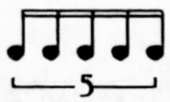
 = 9 nyolcad

- ha a hangjegyet egy pont követi, akkor az eredeti érték a 3/2-szeredsére nő. Pl.:

 = 3 negyed

- nemcsak felezés segítségével lehet hangértéket megadni. Pl. a következő hármas hangcsoport (triola) harmadolja az eredeti értéket.

 = 𝅗𝅥 , tehát mindegyik hangjegy értéke $\frac{1}{2}/3$, azaz $\frac{1}{6}$

Hasonlóan pl.  = ♪ mindegyik jegyének $\frac{1}{8}/5$ azaz $\frac{1}{40}$ az értéke.

- igen gyakran, különösen a kották végén találkozhatunk az ún. koronával: 𝄐 . Ez a jel a hangértéket növeli, a növelés nincs meghatározva, mértékét az előadó választhatja meg.

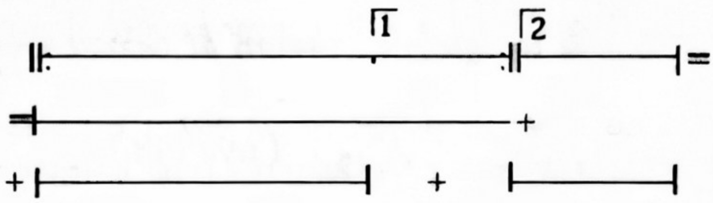
d./ Tempójelölések

Az előzőek alapján a kottából csak a hangértékek egymáshoz való arányát tudjuk megállapítani. Szükség van annak ismertetésére is, hogy az egész értékű hangjegy mekkora időtartamot jelöl. A kották többségének tetején ♩ = 100 alakú jelzés olvasható, mely jelen esetben pl. azt jelenti, hogy egy percre 100 negyedértéknek kell esnie. A tempó pontos mérésére a metronóm szolgál. Az előadásmód ettől eltérő meghatározására olasz szavak szolgálnak, mint pl.:

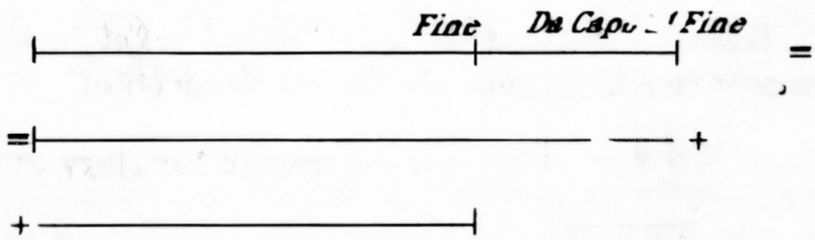
Lento	*nagyon lassan*
Adagio	*lassan*
Andante	*lépve*
Moderato	*mérsékelten*
Allegro	*gyorsan*
Presto	*sebesen*

e./ Ismétlőjelek

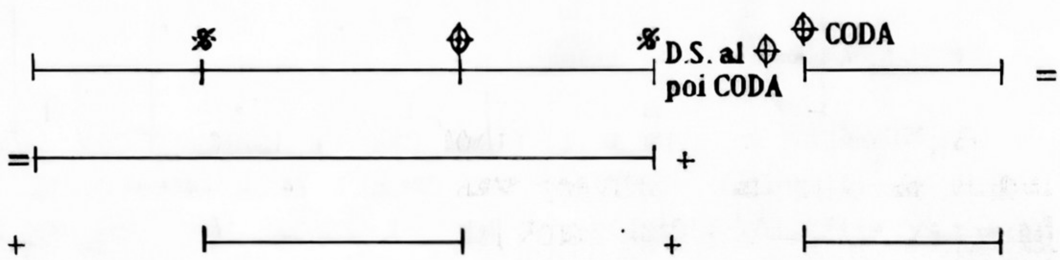
A zeneszámokban gyakran előfordulnak ismétlődő szakaszok. Ezeket is célszerű megjelölnünk, hogy ne kelljen újra és újra lekottáznunk őket. A legegyszerűbb ilyen jel a ||: :||, amely azt mutatja, hogy a kottának a jelpár közötti szakaszát kétszer kell megszólaltatni. Ha egy műben csak a :|| jelet találjuk, akkor ez azt jelenti, hogy az ismétlést a kotta elejéről kell kezdeni. Ha ismétléskor a :|| jel előtti néhány ütemben eltérés lenne, akkor az eltérő rész kezdetét egy |1 jellel jelölik, s egy |2 jel kerül az ezt pótló rész kezdetéhez.



Amennyiben a mű utolsó része megegyezik a kezdetével, akkor a kotta végét a Da Capo (D.C.) al Fine (=elölről a végéig) felirattal látják el, és a mű igazi végét a Fine szóval jelölik.



Előfordulhat, hogy az ismétlés nem elölről történik. Ilyenkor a 𝄋 jel mutatja, hogy honnan kell kezdeni az ismétlést. Popdalokban igen gyakori, hogy a számban megtalálható az előbbi ismétlés, de a szám vége kicsit különbözik az addigiaktól. Ilyenkor a Coda jelzést használják.



f./ Hangerő

A tempó jelöléséhez hasonlóan olasz szavak jelzik a mű előadásának hangerejét is.

pp	pianissimo	= nagyon halkan
p	piano	= halkan
mp	mezzopiano	= középhalkan
mf	mezzoforte	= középerősen
f	forte	= erősen
ff	fortissimo	= nagyon erősen

Az előadás közben szükséges hangerőváltozásokra a következő szemléletes jelek figyelmeztetnek:

<	crescendo (cresc.)	-felerősödve
>	decrescendo (decresc.)	-halkulva
	diminuendo (dim.)	
∧	sforzato (sf.)	=erősen kiemelve

g./ Ütemek

A dallamok többsége egységekre, ütemekre tagolható a hangsúlyok változása szerint. Az ütemeket függőleges vonalak választják el egymástól. Így a kotta jóval áttekinthetőbb, olvasása egyszerűbb lesz. Az ütem beosztását törtszám jelzi (6/4, 3/8...). A tört nevezője az alapul vett hangjegyértéket jelenti (negyed, nyolcad, stb.), a számláló pedig azt mutatja, hogy egy ütemben hány ilyen hangjegyérték szerepel. A 4/4 és a 2/2 ütemet másképp is szokás jelölni: az előbbit C-vel, az utóbbit pedig ₵-vel. Az egyes ütemmutatók egészen addig érvényesek, míg egy új ütemmutatót ki nem teszünk.

Az ebben a fejezetben leírtak ismeretével még nem tekinthetjük magunkat a zeneelmélet professzorainak. Ahhoz azonban elegendőt tudunk, hogy egy átlagos kottát számítógépre vigyünk, s célunk nem is volt több. Akikben sikerült felkelteni az érdeklődést a zeneelmélet iránt, azoknak a [7] könyv tanulmányozását ajánlom.

A SID 6581 chip

A Commodore 64 zenei lehetőségeit, melyek nagyon hasonlóak a Moog szintetizátorokéhoz, a gépbe épített SID chipnek (*sound interface device*) köszönheti. Ez három hang egymástól független kezelését biztosítja. A SID-nek 25 csak írható, és 4 csak olvasható regisztere van. Ebben a fejezetben ezek ismertetésére kerül sor. Üljünk máris a gép elé, és szólaltassuk meg a következő program segítségével!

```
10 S=54272      :REM A SID NULLADIK REGISZTERE
15 POKES+24,15:REM A HANGEROSSEG 15,AZAZ MAX.
20 POKES+6,255:REM 255=16*S+R,IGY S=15 R=15 AZ 1.HANG ESETEBEN
25 POKES+1,20 :REM ITT ALLITJUK BE AZ 1.HANG MAGASSAGAT
30 POKES+4,17 :REM 17=16+1,IGY AZ 1.HANG HULLAMFORMAJA
31 REM            HAROMSZOG ES A KAPU ZARODIK
READY.
```

Az első fontos tudnivaló, hogy a hanggenerálással kapcsolatos memóriarész az S=54272. byte-on kezdődik. Ez a SID chip nulladik regisztere. Ahhoz, hogy egy hang megszólaljon, több dolgot feltétlenül nullától különböző értékűre kell beállítani. Ezek a következők: hangerősség, hangmagasság, a hangdinamika (ADSR) első három paramétere közül legalább az egyik, hullámforma, Gate-bit (kapu). Mielőtt e bűvös szavak jelentéséről többet megtudnánk, futtassuk le ismét a programot!

A program futásakor valóban megszólal egy hang, sőt el sem akar halkulni. Ezt kihasználva kísérletezzünk egy kicsit! Módosítsuk a 15-ös sorban lévő 15-öt 10-re, 5-re, 0-ra. A hang kezd elhalkulni, sőt a végén teljesen elnémul. Ez persze nem véletlen. A SID 24. regiszterének, azaz az S + 24 = 54296. byte alsó négy bitje határozza meg a hangerőt – természetesen a TV, ill. az erősítő hangerőszabályozójával együtt. E hangerő mindhárom hangra vonatkozik. A maximális hangerőt akkor érjük el, ha e négy bit által meghatározott szám 15. Ha pedig egy program futása után még mindig szól a TV, akkor a POKE 54296,0 utasítással némíthatjuk el.

Folytassuk a kísérletezést! A hangerőt állítsuk vissza maximumra (15), majd változtassuk a 25-ös sorban szereplő 20-at 0 és 255 között, illetve a 30-as sorban lévő 17-et a sorrendet betartva 16, 33, 32, 129 majd 128-ra. Minden egyes változtatás után futtassuk le a programot! A változások okait a későbbiekben részletezzük. Ha valaki a kísérletezés során a 20-as sorban szereplő 255-öt is megváltoztatja, akkor előfordulhat, hogy teljesen elmegy a hang, s még az eredeti program visszaállítása után sem szólal meg a TV. Ekkor egy új sorral érdemes bővíteni a programunkat:

```
11 FOR i = 0 TO 24 : POKE S + i, 0 : NEXT
```

E sor hatására a SID chip írható regisztereibe 0 kerül, s így a SID alapállapotba kerül. Mint már tudjuk, a SID chip első 25 regisztere (S-től S+24-ig) nem olvasható, csak írható. Próbáljuk ki az előbbi program futtatása után a következő utasítást!

```
PRINT PEEK (S+1)
```

Eredményül valóban nullát fogunk kapni. Az érdekesség kedvéért próbáljuk ki a következő utasítás-sort is:

```
S = 54272
POKE S,250 : PRINT PEEK (S);PEEK(S);PEEK(S);PEEK(S)
```

vagy ugyanez rövidítve:

```
P┌ S, 250:?P¯(S);P¯(S);P¯(S);P¯(S)
```

A RETURN billentyű megnyomása után vagy a 250, 10, 0, 0 számok, vagy pedig ehhez közeli értékek kerülnek kiírásra.

A továbbiak jobb megértése érdekében tekintsük át a zenei hanggal kapcsolatos legfontosabb tudnivalókat! A hangrezgés legfontosabb tulajdonságai

a hangerősség,
a hangmagasság,
a hangszín és
a hang dinamikája.

2.1 A hangerősség

A hang erősségéről kétféle értelemben beszélhetünk:

a) fizikai értelemben a hangerősséget hangintenzitással jellemezzük:

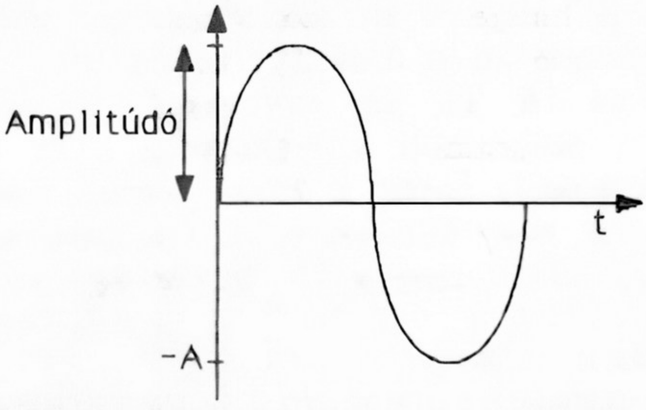
$$I=\frac{1}{2}\varrho c v^2_{max}.$$

ahol ϱ annak a közegnek a sűrűsége, amelyben a hang terjed
c a hang terjedési sebessége e közegben
v_{max} a közeg részecskéinek maximális rezgési sebessége.

Mértékszáma egyenlő a terjedés irányára merőleges felületegységen az időegység alatt átsugárzott energia mértékszámával.

Egysége: $1 \frac{J}{s\,m^2} = 1 \frac{W}{m^2}$

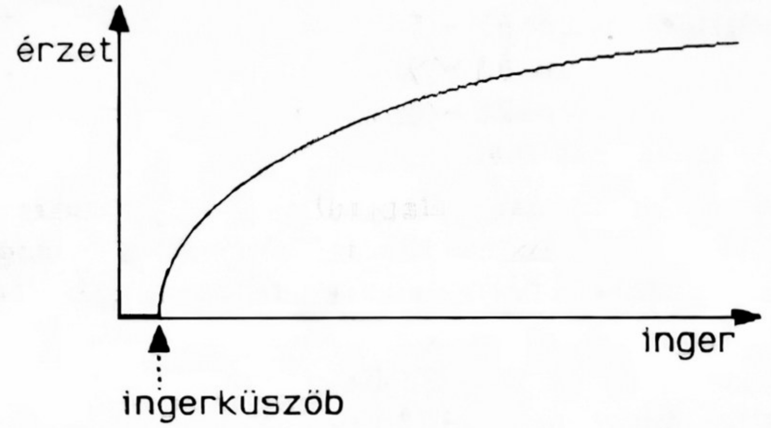
Ha a hang levegőben terjed, akkor a hangerősséget a hangrezgés amplitúdója határozza meg. Pl. szinuszos rezgések esetén az amplitúdó a rezgést leíró függvény maximuma.



7. ábra

b) fiziológiai értelemben a hangerősség az érzékelt "hangosság", melyet a fülbe érkező hang nyomásingadozása kelt. A rezgés amplitúdójának növelésével a hangerősség érzete is nő, de nem egyenes arányban. Az, ahogy a fül a hangintenzitást, a hangnyomás amplitúdóját érzékeli, egy logaritmikus függvénnyel írható le. Az érzet

erőssége tehát az ingernél lényegesen lassabban nő. A kettő közti kapcsolatot jól mutatja a következő ábra:



8. ábra

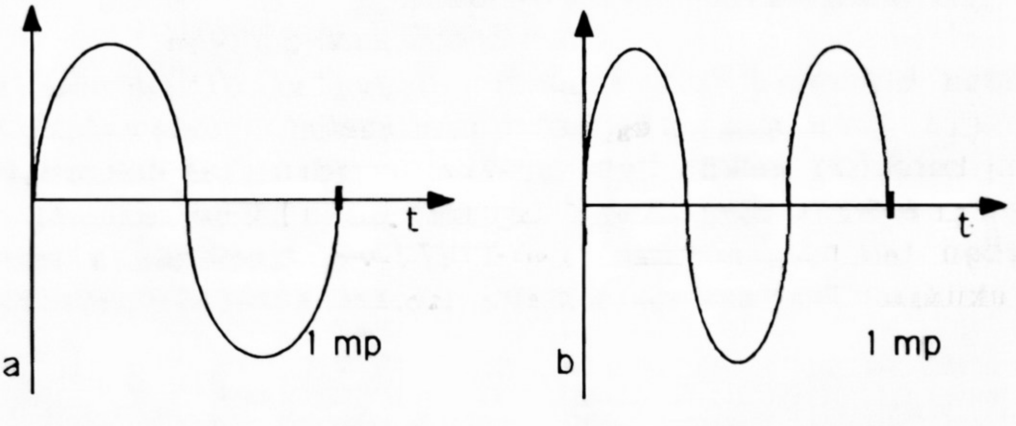
A fiziológiai értelemben vett hangerősség mértékegysége az 1 phon. Két adat ennek érzékeltetésére:

suttogás	kb.10 phon
repülőgép közvetlen közelről	kb 120 phon

Mint már tudjuk, a C-64-en az S+24=54296. byte alsó négy bitje határozza meg a hangerőt.

2.2 A hangmagasság

A hangmagasság az alaprezgés frekvenciájától függ. Minél nagyobb a frekvencia (azaz a másodpercenkénti rezgésszám), annál nagyob a hang magassága. A frekvencia mértékegysége az 1 Hz, amely Heinrich Hertz német fizikusról kapta nevét. (A 9/a ábrán látható szinuszhullám frekvenciája 1 Hz, míg a 9/b ábrán 1,5 Hz.)



9. ábra

Az emberi fül által hallható frekvencia 16 és 20000 Hz között van. A 20 Hz alatti rezgéseket infrahangnak, a 16000 Hz-nél nagyobb frekvenciájúakat ultrahangnak, míg a 10^8 Hz = 100 megahertznél nagyobbakat hiperhangnak nevezik.

A Commodore 64 hangterjedelme 0 – kb. 4000 Hz. Ez kevésnek tűnhet, de ha figyelembe vesszük, hogy

a gitár hangterjedelme	kb. 82 – 987 Hz,
a csellóé	kb. 65 – 900 Hz,
a zongoráé	kb. 28 – 4000 Hz,

akkor a C-64 hangterjedelme is jónak mondható.

A hangszerek egységes hangolásának alapjául az egyvonalas "a" hang frekvenciáját határozták meg. Ezt a hangot szokás "normál a" hangnak, vagy "kamarahang"-nak is hívni. Érdekes, hogy ennek frekvenciáját többször is módosították.

1789 bécsi hangolás	421 Hz
1850 bécsi és berlini hangolás	442 Hz
1858 párizsi konferencia	435 Hz
1939 nemzetközi megállapodás	440 Hz

Ez utóbbi a mai napig elfogadott változat.

A C-64-en a hangmagasságot a következőképpen lehet beállítani:

(mindhárom hang számára 2-2 byte áll rendelkezésre ilyen célra)

az 1. hang számára az S	- 54272	és az S+1	- 54273
a 2. hang számára az S+7	- 54279	és az S+8	- 54280
a 3. hang számára az S+14	- 54286	és az S+15	- 54287

Két byte-ot egymás után írva egy 16 bites számot kapunk. Tegyük meg ezt a fenti byte-párok esetében úgy, hogy a felső byte rendre az S+1., S+8., ill. az S+15. regiszter értéke legyen. Az így kapott P szám értéke pl. az 1. hang esetében P=256·PEEK(S+1)+PEEK(S).

P-t a megszólaltatni kívánt hang F frekvenciájából a P=F/k összefüggés alapján határozhatjuk meg, ahol k egy konstans. Ennek értékére a különböző irodalmak eltérő értékeket közölnek: [13]-ban és [14]-ben k=0,06097,
[1]-ben k=0,06097 és k=0,05966

Az utóbbi azonban bizonyára csak sajtóhiba, hiszen az itt szereplő adatok többségükben a [2]-ben közölt adatokkal egyeznek meg; ezeket viszont k=0,06096-tal számították, ráadásul kerekítés nélkül. Ugyanezekkel az adatokkal dolgoznak a [4] könyv szerzői, és ezek az értékek alkotják az 1. táblázat utolsó három oszlopát.

A [3] függelékében található adatokat k=0,05872-vel számolták a szerzők s szerencsére már kerekítéssel. Ezek szerepelnek az 1. táblázat középső harmadában.

	HANG	(HZ)	PARA-METER	F.B	A.B	P2	F2	A2
0	C-0	16.4	278	1	22	268	1	12
1	C#-0	17.3	295	1	39	284	1	28
2	D-0	18.4	313	1	57	301	1	45
3	D#-0	19.4	331	1	75	318	1	62
4	E-0	20.6	351	1	95	337	1	81
5	F-0	21.8	372	1	116	358	1	102
6	F#-0	23.1	394	1	138	379	1	123
7	G-0	24.5	417	1	161	401	1	145
8	G#-0	26	442	1	186	425	1	169
9	A-0	27.5	468	1	212	451	1	195
10	A#-0	29.1	496	1	240	477	1	221
11	H-0	30.9	526	2	14	506	1	250
12	C-1	32.7	557	2	45	536	2	24
13	C#-1	34.6	590	2	78	568	2	56
14	D-1	36.7	625	2	113	602	2	90
15	D#-1	38.9	662	2	150	637	2	125
16	E-1	41.2	702	2	190	675	2	163
17	F-1	43.7	743	2	231	716	2	204
18	F#-1	46.2	788	3	20	758	2	246
19	G-1	49	834	3	66	803	3	35
20	G#-1	51.9	884	3	116	851	3	83
21	A-1	55	937	3	169	902	3	134
22	A#-1	58.3	992	3	224	955	3	187
23	H-1	61.7	1051	4	27	1012	3	244
24	C-2	65.4	1114	4	90	1072	4	48
25	C#-2	69.3	1180	4	156	1136	4	112
26	D-2	73.4	1250	4	226	1204	4	180
27	D#-2	77.8	1325	5	45	1275	4	251
28	E-2	82.4	1403	5	123	1351	5	71
29	F-2	87.3	1487	5	207	1432	5	152
30	F#-2	92.5	1575	6	39	1517	5	237
31	G-2	98	1669	6	133	1607	6	71
32	G#-2	103.8	1768	6	232	1703	6	167
33	A-2	110	1873	7	81	1804	7	12
34	A#-2	116.5	1985	7	193	1911	7	119
35	H-2	123.5	2103	8	55	2025	7	233
36	C-3	130.8	2228	8	180	2145	8	97
37	C#-3	138.6	2360	9	56	2273	8	225
38	D-3	146.8	2500	9	196	2408	9	104
39	D#-3	155.6	2649	10	89	2551	9	247
40	E-3	164.8	2807	10	247	2703	10	143
41	F-3	174.6	2974	11	158	2864	11	48
42	F#-3	185	3150	12	78	3034	11	218
43	G-3	196	3329	13	10	3215	12	143
44	G#-3	207.7	3536	13	208	3406	13	78
45	A-3	220	3746	14	162	3608	14	24
46	A#-3	233.1	3969	15	129	3823	14	239
47	H-3	246.9	4205	16	109	4050	15	210
48	C-4	261.6	4455	17	103	4291	16	195
49	C#-4	277.2	4720	18	112	4547	17	195
50	D-4	293.7	5001	19	137	4817	18	209
51	D#-4	311.1	5298	20	178	5103	19	239
52	E-4	329.6	5613	21	237	5407	21	31
53	F-4	349.2	5947	23	59	5728	22	96
54	F#-4	370	6301	24	157	6069	23	181

1. táblázat

	HANG	(HZ)	PARA-METER	F.B	A.B	P2	F2	A2
55	G-4	392	6676	26	20	6430	25	30
56	G#-4	415.3	7072	27	160	6812	26	156
57	A-4	440	7493	29	69	7217	28	49
58	A#-4	466.2	7939	31	3	7647	29	223
59	H-4	493.9	8411	32	219	8101	31	165
60	C-5	523.3	8911	34	207	8583	33	135
61	C#-5	554.4	9441	36	225	9094	35	134
62	D-5	587.3	10002	39	18	9634	37	162
63	D#-5	622.3	10597	41	101	10207	39	223
64	E-5	659.3	11227	43	219	10814	42	62
65	F-5	698.5	11894	46	118	11457	44	193
66	F#-5	740	12602	49	58	12139	47	107
67	G-5	784	13351	52	39	12860	50	60
68	G#-5	830.6	14145	55	65	13625	53	57
69	A-5	880	14986	58	138	14435	56	99
70	A#-5	932.3	15877	62	5	15294	59	190
71	H-5	987.8	16821	65	181	16203	63	75
72	C-6	1046.5	17821	69	157	17167	67	15
73	C#-6	1108.7	18881	73	193	18188	71	12
74	D-6	1174.7	20004	78	36	19269	75	69
75	D#-6	1244.5	21193	82	201	20415	79	191
76	E-6	1318.5	22454	87	182	21629	84	125
77	F-6	1396.9	23789	92	237	22915	89	131
78.	F#-6	1480	25203	98	115	24278	94	214
79	G-6	1568	26702	104	78	25721	100	121
80	G#-6	1661.2	28290	110	130	27251	106	115
81	A-6	1760	29972	117	20	28871	112	199
82	A#-6	1864.7	31754	124	10	30588	119	124
83	H-6	1975.5	33642	131	106	32407	126	151
84	C-7	2093	35643	139	59	34334	134	30
85	C#-7	2217.5	37762	147	130	36376	142	24
86	D-7	2349.3	40008	156	72	38539	150	139
87	D#-7	2489	42387	165	147	40830	159	126
88	E-7	2637	44907	175	107	43258	168	250
89	F-7	2793.8	47578	185	218	45830	179	6
90	F#-7	2960	50407	196	231	48556	189	172
91	G-7	3136	53404	208	156	51443	200	243
92	G#-7	3322.4	56580	221	4	54502	212	230
93	A-7	3520	59944	234	40	57743	225	143
94	A#-7	3729.3	63508	248	20	61176	238	248
95	H-7	3951.1	67285	262	213	64814	253	46

Ebben a táblázatban

az 1. oszlop a sorszám,
a 2. oszlop a hang neve,
a 3. oszlop a hang F frekvenciája Hz-ben,
a 4. oszlop a [3]-ban használt P=F/k paraméter,
az 5. ill. 6. oszlop az előbbi szerinti felső, ill. alsó byte értéke,
a 7. oszlop a [2]-ben használt P=F/k paraméter,
a 8. ill. 9. oszlop az utóbbi szerinti felső ill. alsó byte értéke

MFeladat 2.

Készítsünk frekvenciatáblázatunkhoz hasonló táblázatkészítő programot!

Felmerülhet a kérdés: érdekes-e egyáltalán, hogy k=0,06097-tel, vagy k=0,05872-vel számolunk, hiszen e két érték között a különbség csak 0,00225. Ez azonban csalóka, hiszen a k-val osztani kell. Ha elvégezzük a P=F·(1/k) átalakítást, akkor az első esetben P=F·16,4015, a második esetben pedig P= F·17,02997 adódik, és így a két érték közötti nagy eltérés már nyilvánvaló lesz. Nézzük meg az 1. táblázat megfelelő adatait is.

A következő program a táblázatban szereplő két különböző A-4 hangot váltogatja:

```
10 S=54272                 :REM A SID KEZDOCIME
15 FORI=0TO24:POKES+I,0:NEXT:REM KINULLAZZUK A SID IRHATO
16 REM                          REGISZTEREIT
20 POKES+24,15             :REM A HANGERO MAX.
25 POKES+5,15              :REM 15=16*A+D IGY A=0 D=15 AZ 1.
26 REM                          HANG ESETEBEN
30 POKES+4,17              :REM AZ 1.HANG HULLAMFORMAJA
31 REM                          HAROMSZOG ES A KAPU ZARVA
35 FORJ=1TO10              :REM 10-SZER VALTOZTATJUK A "KET"
36 REM                          A-4 -ES HANGOT
40 POKES+1,29:POKES,69:REM AZ 1.HANG HANGMAGASSAGA
41 REM                          A-4 /K=0.05872 ESETEN/
45 FORI=1TO500:NEXTI       :REM A MEGSZOLALO HANG KITARTASA
50 POKES+1,28:POKES,49:REM AZ 1.HANG HANGMAGASSAGA
51 REM                          A-4 /K=0.06096 ESETEN/
55 FORI=1TO500:NEXTI       :REM A MODOSITOTT HANG KITARTASA
60 NEXTJ
```

A két hang közötti különbség bizonyára jól hallható, s így nem pusztán elméleti jelentőségű annak eldöntése, hogy melyik érték az "igazi". Az a legbiztosabb módszer, hogy egy hangvillával ellenőrizzük a különböző paraméterek segítségével előállított hangok helyességét. (Hangvilla hiányában hangsíppal vagy pl. jobb minőségű furulyával is tesztelhetjük a hangot.) A tesztelés során azonnal kitűnik, hogy bár az említett könyvek többségében a szerzők k=0,06097-tel számoltak, a (28,49) paraméterpárral előállított A-4 alacsonyabb a "normál a"-nál. Ezzel szemben a (29,69) paraméterpárral előállított hang lényegében megfelelő. Ennél pontosabban csak műszerrel lehetne megállapítani a valódi k értéket, és az ennek megfelelő paramétereket, ennek azonban tényleg csak elméleti jelentősége lenne. Tegyük fel pl. hogy a "normál a" hangot a (29,60) paraméterpárral lehet megadni. Módosítsuk az előbbi program 50-es sorában levő (28,49)-et (29,60)-ra. Futtassuk le módosított programunkat!

Annak, aki most is két különböző hangot hall, gratulálok, mert tényleg jó füle van!

A számítógép hangolásának csak akkor van igazán fontos szerepe, ha az más hangszerrel játszik együtt, tehát pl. a C-64-re írt zenét furulyával vagy zongorával akarjuk kísérni. (Húros hangszerek esetén nincs probléma, mert azokat a C-64-hez tudjuk hangolni.) Ha azonban nem vágyunk magánzenekar alapítására, akkor teljesen mindegy, hogy a k említett értékei közül melyiket fogadjuk el, csak az a fontos, hogy egy programon belül következetesen mindig ugyanazzal a k-val számítsuk a további paramétereket.

A biztonság kedvéért nézzünk egy konkrét példát adott frekvencia programozására. Ha pl. a harmadik hang frekvenciáját 200 Hz nagyságúra akarjuk beállítani, akkor a következőket kell tennünk:

1. Kiszámítjuk a P paraméter értékét:

$$P = F/k = 200/0{,}05872 \approx 3406$$

2. Kiszámítjuk a felső és az alsó byte értékét

$$FB = INT(P/256) = INT(3406/256) = 13$$
$$AB = P-256 \cdot FB = 3406-256 \cdot 13 = 78$$

3. A byte-értékeket beírjuk a SID megfelelő regisztereibe. Most a harmadik hangról lévén szó

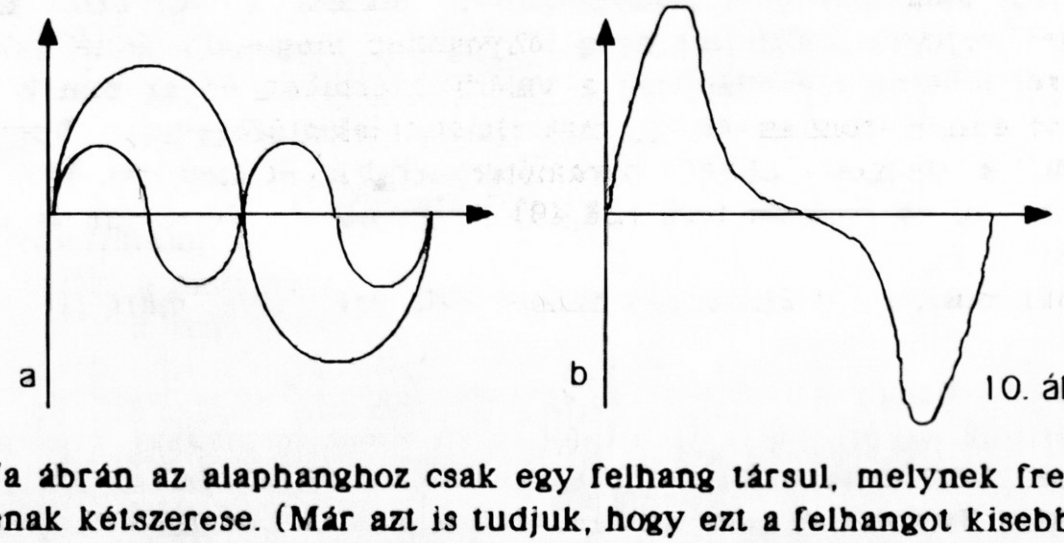
POKE S+15, FB azaz POKE S+15,13 és
POKE S+14, AB azaz POKE S+14,78
ahol S = 54272

2.3 A hangszín

A szaxofonon megfújt hangok egészen másképpen szólnak, mint pl. a zongorán leütött azonos magasságú, frekvenciájú hangok. Ez annak köszönhető, hogy az alaphang megszólalásakor az ehhez csatlakozó felhangok, felharmonikusok is megszólalnak. Egy alaphang felhangjainak azokat a hangokat nevezzük, amelyeknek frekvenciái az alaphang frekvenciájának egész számú többszörösei.

A 440 Hz frekvenciájú A-4 felhangjai például a 880 Hz frekvenciájú A-5, az 1320 Hz frekvenciájú hang, az 1760 Hz frekvenciájú A-6, és így tovább.

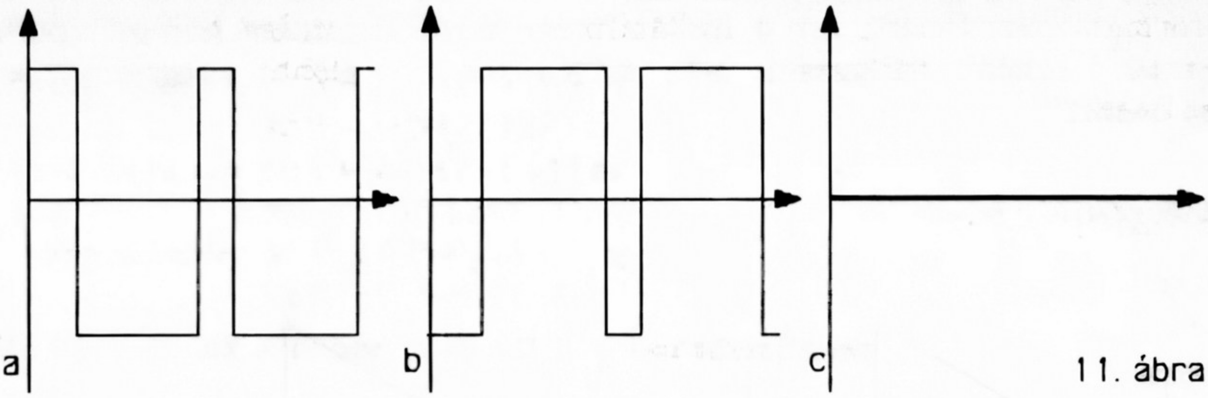
A hang színezetét – Helmholtz törvénye alapján – az alaphanghoz csatlakozó felhangok frekvenciája és viszonylagos erőssége, vagyis a hang rezgési spektruma határozza meg.



10. ábra

A 10/a ábrán az alaphanghoz csak egy felhang társul, melynek frekvenciája az alaphangénak kétszerese. (Már azt is tudjuk, hogy ezt a felhangot kisebb hangerővel halljuk, mint az alaphangot, hiszen kisebb az amplitúdója.) Ebben az esetben a ténylegesen megszólaló hang rezgését a 10/b ábrán látható grafikonnal adhatjuk meg, hiszen kettő vagy több hullám eredőjét úgy kaphatjuk meg, hogy egyszerűen összeadjuk őket. Így pl. azok a hanghullámok, melyeket a 11/a és b ábra görbéi

határoznak meg, külön-külön hallhatóak lennének, együtt azonban nem, mert kioltják egymást. (11/c ábra)



11. ábra

Ezt a kísérletet később a gép segítségével is el fogjuk végezni.

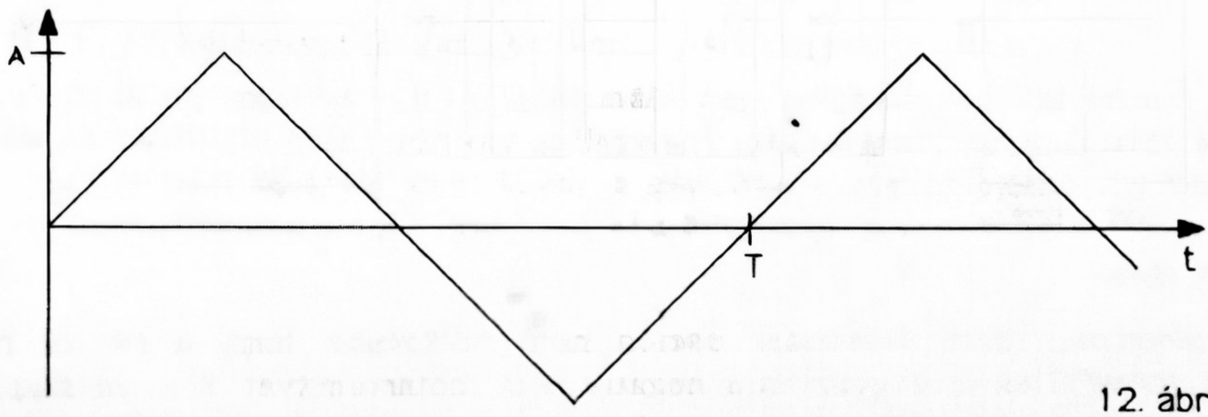
Itt az ideje, hogy megismerkedjünk a C-64, azaz a SID chip által biztosított hullámformákkal és azok programozásával. Mindhárom hanghoz egy ún. kontroll regiszter tartozik. Ezek a következők:

az 1. hang esetén	S+4	= 54276
a 2. hang esetén	S+11	= 54283
a 3. hang esetén	S+18	= 54290

A következőkben a C-64-be "beépített" 4 hullámformát fogjuk megismerni, melyeket úgy tudunk kiválasztani az egyes hangok számára, hogy az említett kontroll-regiszterek következő bitjeit magasra állítjuk:

Hullámforma	Bit
háromszög	4.
fűrészfog	5.
négyszög	6.
fehér zaj	7.

a) *Háromszög hullám*

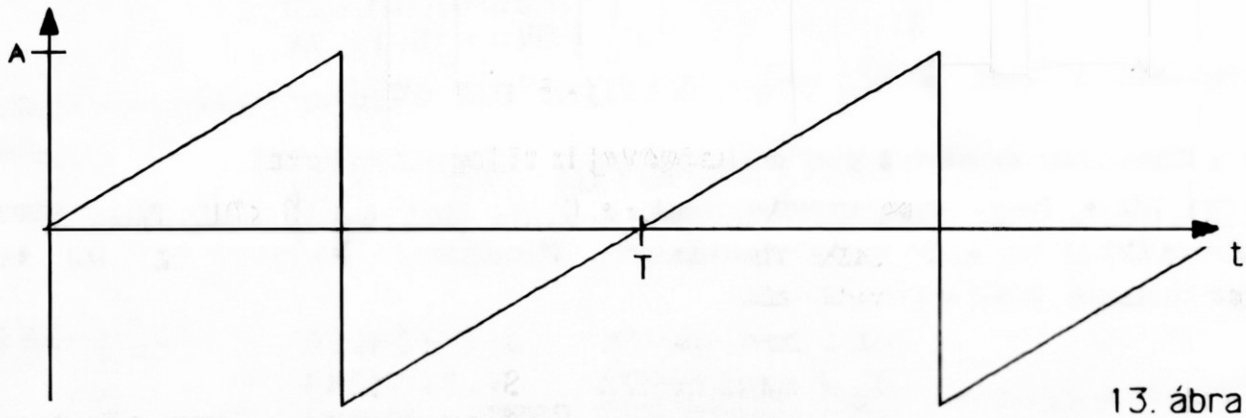


12. ábra

A háromszöghullám alaphangjához csak azok a felhangok csatlakoznak, melyeknek frekvenciája az alaphang frekvenciájának páratlan számú többszöröse. Hogy ez miért van így, ill. hogy a többi hullámformához milyen felhangok csatlakoznak, arról a matematika iránt érdeklődők az 1. Függelékben olvashatnak. Mivel ez a hullámforma igen hasonlít a szinuszjelhez, ezért lágy hangzású, leginkább

talán a fuvola hangjához hasonlítható. Az A-4 hangot előállító programban háromszög hullámformát használtunk, így e hullámforma további leírása helyett (puha, lágy, selymes stb.) inkább hallgassuk meg mégegyszer az előbbi programot, az úgyis magáért beszél.

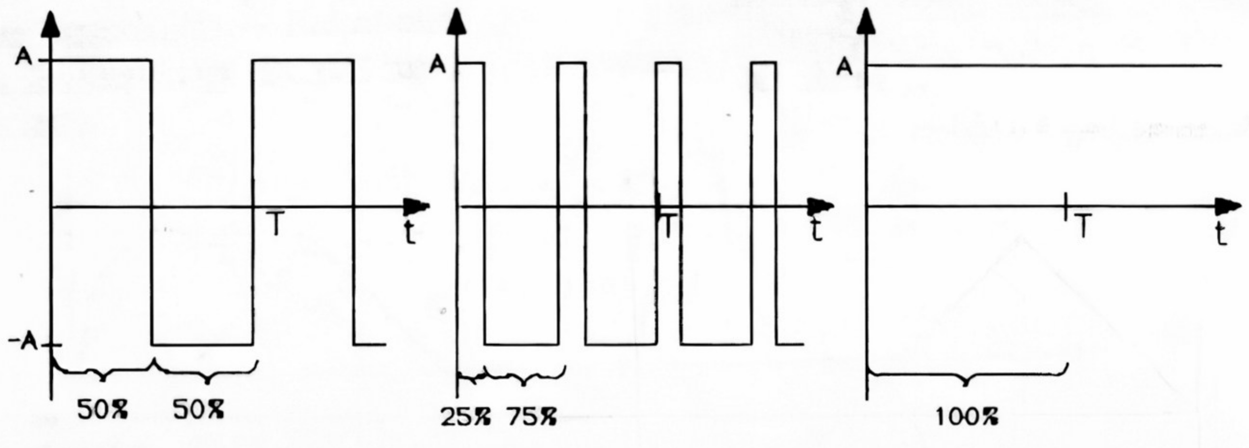
b) Fűrészfoghullám



13. ábra

A hullámforma által előállított hangzás leginkább a vonós hangszerek hangjához hasonlítható. Ha az előbbi program 30-as sorában a POKE S + 4,17 utasítást POKE S + 4,33-ra módosítjuk, akkor fűrészfoghullám lesz az új hullámforma.

c) Négyszöghullám



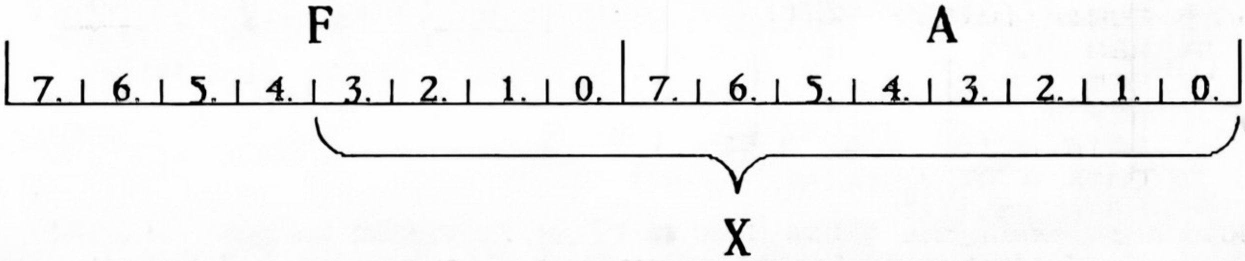
14. ábra a b c

A négyszöghullám választása esetén nem szükséges, hogy a rezgés pozitív részének időtartama megegyezzen a negatív rész időtartamával. E kettő százalékos arányát, az ún impulzusszélességet is módosíthatjuk. A 14/b ábrán például egy 25%-os négyszög-rezgés grafikonja látható. Jegyezzük meg, hogy a 25% a pozitív rész arányát jelöli a T periódus időből.

Az arány programozására a C-64-en minden hang számára 2-2 byte, azaz a SID chip 2-2 regisztere áll rendelkezésre.

Az I. hang számára az S + 2=54274 (A) és
az S + 3=54275 (F),
a II. hang számára az S + 9=54281 (A) és
az S +10=54282 (F),
a III. hang számára az S +16=54288 (A) és
az S +17=54289 (F).

A felső byte-oknak azonban csak a 0-3. bitjei számítanak.



Az így kapott 12 bites szám értékét úgy kell meghatározni, hogy a kívánt százalékos arány 40,95-szorosa legyen. (Ez a szám azért éppen 40,95, mert a 100%-hoz az $111111111111_2=2^{12}-1$ = 4095 tartozik, következésképp x%-hoz az $X = x \cdot 40{,}95$ ahol $0 \leq x \leq 100$.)

Ezek szerint ahhoz például, hogy a 14/b ábrán látható 25%-os négyszög hullámforma legyen a II. hang hullámformája, a következő utasításokat kell kiadnunk:

```
(S = 54272)
POKE S + 9, 0
POKE S + 10,4
```

$(X = 25 \cdot 40{,}95 \approx 1024)$

A POKE S + 11,65 hatására a II. hang hullámformája négyszöghullám lesz.

Ne felejtsük el, hogy ha egy hang számára négyszöghullámot választunk, akkor mindig be kell állítanunk az impulzus szélességét meghatározó két regiszter értékét is, mert, ha ezt nem tesszük meg, akkor a gép 0%-os négyszögjellel fog dolgozni, amely nem hallható konstans jel. Próbáljuk ki a következő programot!

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT:REM A SID REGISZTEREINEK
16 REM                         TORLESE
20 POKES+24,15               :REM HANGEROSSEG MAX.
25 POKES+12,15               :REM A=0 D=15 A 2.HANG ESTEBEN
30 POKES+11,65               :REM 65=64+1 IGY A 2.HANG
31 REM                            HULLAMFORMAJA NEGYSZOG
32 REM                            ES A KAPU ZARVA
35 FORJ=1TO11
40 READXF,XA                 :REM AZ IMPULZUSSZELESSEGET
41 REM                            MEGHATAROZO ADATOK BEOLVASASA
45 POKES+10,XF:POKES+9,XA:REM A 2.HANG IMP.SZELESSEGENEK
46 REM                            A BEALLITASA
50 POKES+8,7 :POKES+7,81  :REM A 2.HANG HANGMAGASSAGA A-2
55 FORI=1TO1000:NEXTI     :REM EZ AZ URESCIKLUS HATAROZZA
56 REM                            MEG,HOGY AZ EGYES HANGOK
57 REM                            MENNYI IDEIG SZOLJANAK
60 NEXTJ
65 DATA 0,19, 0,0, 0,62, 1,0, 2,0
70 DATA 4,0, 6,0, 8,0, 12,0, 15,0
75 DATA 15,100
```

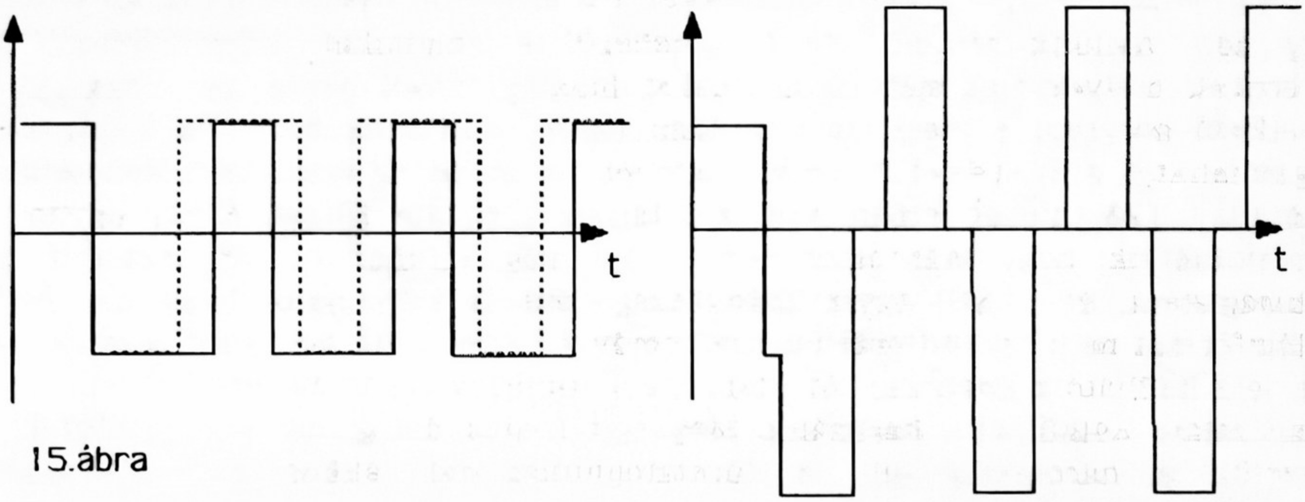
A program futásakor különböző négyszög-hullámformájú A-2 hangot hallunk a második esetet kivéve, amikor az impulzusszélesség nulla.

A következő programban egy kb. 6,25%-os, majd egy 93,75%-os hangot szólaltatunk meg. A 11/a és b ábra kapcsán elmondottak után azt várhatnánk, hogy a második hang bekapcsolásakor a két hang kioltja egymást, tehát nem hallunk semmit. Ha jobban átgondoljuk, hogy mi is történik valójában, rájövünk, hogy hol tévedtünk. Előbb azonban futtassuk le egymás után többször a programot!

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,10                :REM HANGEROSSEG 10
25 POKES+5,13:POKES+12,13:REM A=0 D=13 AZ 1. ES A 2. HANG
26 REM                             ESETEBEN IS
30 POKES+3,1                  :REM AZ 1.HANG IMP.SZELESSEGENEK
31 REM                             BEALLITASA
35 POKES+10,14:POKES+9,255:REM A 2.HANG IMP.SZELESSEGENEK
36 REM                             BEALLITASA
40 POKES+4,65:POKES+11,65:REM AZ 1. ES A 2.HANG HULLAM-
41 REM                             FORMAJA IS NEGYSZOGHULLAM
42 REM                             A KAPUK BEZARVA
45 POKES+1,7  :POKES,81     :REM AZ 1.HANG HANGMAGASSAGA A-2
46 REM                             ITT SZOLAL MEG AZ 1.HANG
50 FORI=1TO500:NEXTI
55 POKES+8,7  :POKES+7,81   :REM A 2.HANG HANGMAGASSAGA A-2
56 REM                             ITT SZOLAL MEG A 2.HANG
57 REM                             DE HA SZERENCSENK VAN,AKKOR
58 REM                             A KET HANG KIOLTJA EGYMAST
```

Bár a második hang bekapcsolásakor legtöbbször továbbra is hallunk egy A-2 hangot, de ennek hangszíne más és más lesz. Ha elég sokat próbálkozunk, akkor néha valóban előfordul, hogy nem hallunk semmit. A magyarázat egyszerű. A 11/a és b

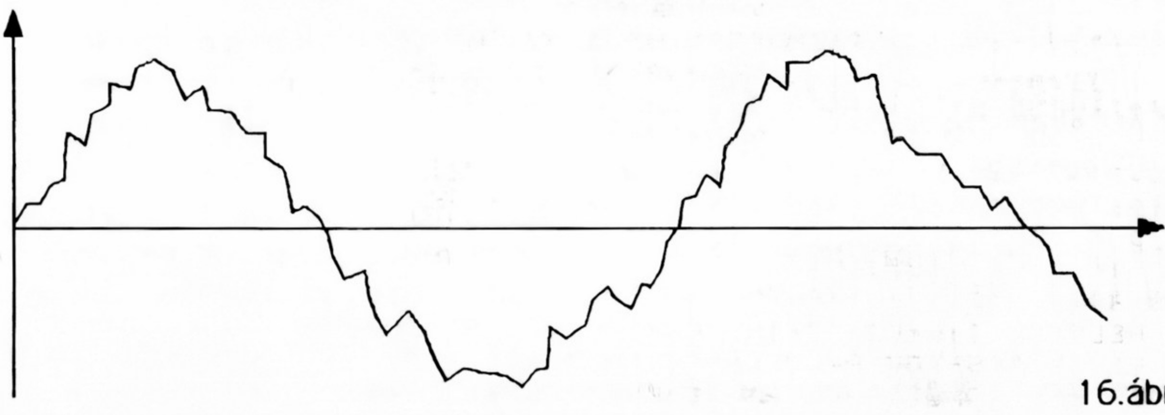
ábrán szereplő hangok egyazon időpontban kezdenek új periódust, míg esetünkben ez nincs feltétlenül így.



15.ábra

Próbáljuk ki, hogy ha elhagyjuk az 50-es sort, akkor megoldódik-e a probléma. Az eredmény lényegében változatlan, azzal a különbséggel, hogy most az 1. hangot önállóan már szinte nem is halljuk. Érdemes azon elgondolkozni, hogy bár a program elején nullára állítjuk a SID írható regisztereinek tartalmát, és utána mindig ugyanazokat a paraméterértékeket állítjuk be, ennek ellenére a megszólaló hang – amely tulajdonképpen két hang eredője – minden futtatáskor másképpen szól. Ennek az az oka, hogy a programon belül megadott BASIC utasításokat az ismételt futtatások esetén nem pontosan ugyanannyi idő alatt hajtja végre a gép.

d) Fehér zaj hullám



16.ábra

A zenei hangok és a zörejek, dörejek közti különbség fő oka az, hogy amíg a zenei hangok lényegében periodikus rezgések, addig a zörejek nem periodikus, szabálytalan rezgések.

Mint már említettük, ezt a hullámformát úgy állíthatjuk elő, hogy a megfelelő hanghoz tartozó kontroll-regiszterbe 129-et írunk. Az első hang esetében például POKE S+4,129 (ahol S=54272). Próbáljuk ki ennek a hangzását is, valamelyik eddigi program módosításával! Ezt a hullámformát jól alkalmazhatjuk majd dörrenések, lövések, kártyaosztás, stb. hangjának szimulálására.

e) kevert hullámformák

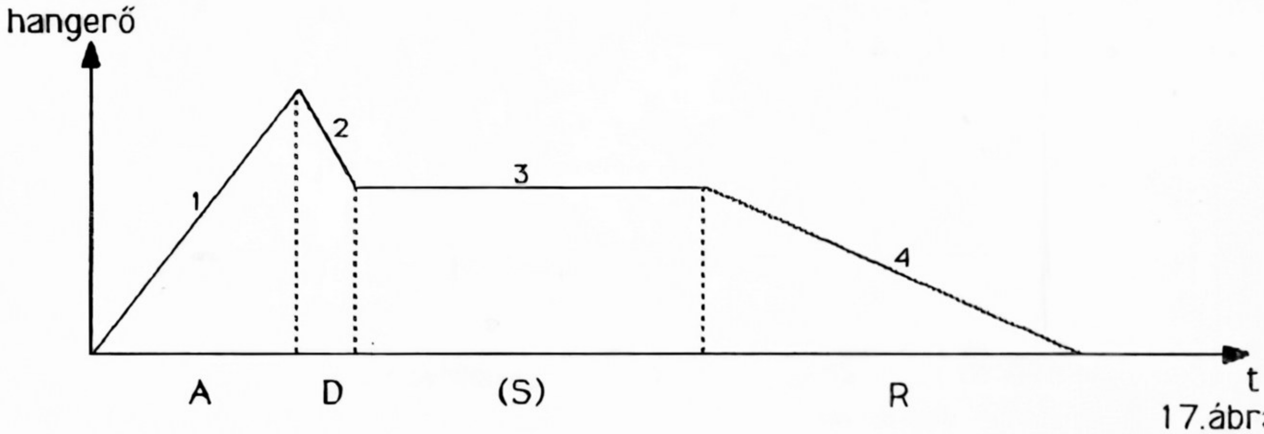
Az említett négy hullámformát keverni is lehet, de ilyenkor gyakran előfordul, hogy nem hallunk semmit. Tovább nehezíti a dolgunkat, hogy vannak olyan keverékek, melyek csak mély hangokkal szólnak jól, mások pedig csak magasakkal. A következő program a megismert hullámformák után azok bizonyos keverékeit is megszólaltatja, a HF=144=128+16 kivételével. Az utolsó HF érték nem véletlenül lett ismételten 128, megegyezően a 4. adattal. A program futása során ugyanis azt tapasztalhatjuk, hogy másodszor nem szólal meg a fehér zaj. Programunk egyik tanulságaként, és a SID egyik hiányosságaként is felfogható, hogy a fehér zaj hullámformát nem szabad más hullámformával keverni. Ha mégis ezt tesszük, akkor nem lesz hallható a keverék, sőt utána nem tudjuk a fehér zaj hullámformát a gép kikapcsolása nélkül újra használni. Még egy fontos dolog: ha a négyszöghullámot keverjük a háromszög- ill. a fűrészfoghullámmal, akkor a négyszöghullám impulzusszélessége lehet 0 is.

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,5          :REM HANGERO 5
25 POKES+6,255         :REM 255=16*D+R,IGY D=15 R=15
26 REM                      AZ 1.HANG ESETEBEN
30 POKES+3,1           :REM AZ 1.HANG IMP.SZELESSEGENEK
31 REM                      BEALLITASA
35 FORK=1TO10          :REM 10 ADATOT FOGUNK BEOLVASNI
40 IFK=5 THEN POKES+24,15:REM KEVERT HULLAMFORMAKNAL
41 REM                           MAXIMALISRA NOVELJUK A
42 REM                           HANGEROT
45 READ HF:PRINT HF    :REM A HULLAMFORMAT MEGHATAROZO
46 REM                      ERTEK BEOLVASASA ES KEPERNYORE
47 REM                      VALO KINYOMTATASA
50 POKES+4,HF+1        :REM AZ 1.HANG HULLAMFORMAJANAK
51 REM                      MEGADASA,A KAPU ZARASA
55 FORJ=1TO250 STEP10:REM J A HANGMAGASSAGOT FOGJA
56 REM                      MEGHATAROZNI
60 POKES+1,J
65 FORI=1TO200:NEXTI   :REM EGY-EGY HANG KITARTASA
70 NEXTJ,K
75 DATA 16,32,64,128,144
80 DATA 48,80,96,112,128
90 REM PELDAUL 112=64+32+16,TEHAT ENNEK AZ ADATNAK A
91 REM BEOLVASASAKOR A HULLAMFORMA A NEGYSZOG-,A FURESZFOG-
92 REM ES A HAROMSZOGHULLAM KEVEREKE LESZ
```

Ha végignézzük előbbi programunk listáját, csak a 25-ös sor szerepe tisztázatlan. A következőkben ez is világossá fog válni. A hullámformák keverésének tanulmányozását sokban segíti majd a szintetizátor, melyet a 4. fejezetben készítünk el.

2.4 A hang dinamikája

A hang dinamikáját egy burkológörbe (ADSR) generátor révén lehet beprogramozni.



17.ábra

Egy hangszeren megszólaló hang hangereje szinte állandóan változik. A megszólalás pillanatától kezdve a hang teljes elhalkulásáig a következő fázisokat különböztethetjük meg:

a hangerő (1) felfut egy maximumig (*attack*)
(2) lecseng egy közbenső szintig (*decay*)
(3) ezen a szinten esetleg tovább kitart (*sustain*)
(4) fokozatosan vagy hirtelen elhalkul (*release*)

E négy fázis angol neveinek kezdőbetűi alapján szokás ezt a generátort ADSR-generátornak is nevezni.

A figyelmes Olvasó most bizonyára azt kérdezi, hogy milyen hangerősségi szintig cseng le a 2. fázisban a hang. A kérdés teljesen jogos. A megoldás C-64 esetében a következő: Az A, D, R-rel ellentétben, melyek időtartamot jelentenek, az S nem a kitartás idejét, hanem a közbenső és a maximális szint arányát határozza meg. (A maximális szintet, mint már tudjuk, a SID 24. regisztere adja meg.) A közbülső szintet elvileg korlátlan ideig kitarthatjuk az ún. GATE-bit (magyarul kapu-bit) magasra állításával. Ameddig a GATE-bit értéke 1, azaz a kapu zárva van, addig a hang "kénytelen" a közbülső szinten várakozni, de amint kinyílik a kapu, azaz a GATE-bit 0 lesz, elkezdődik a 4. fázis, a hang teljes elhalkulása. A GATE-bit a kontroll regiszter (S+4; S+11 ill. S+18) 0. bitje.

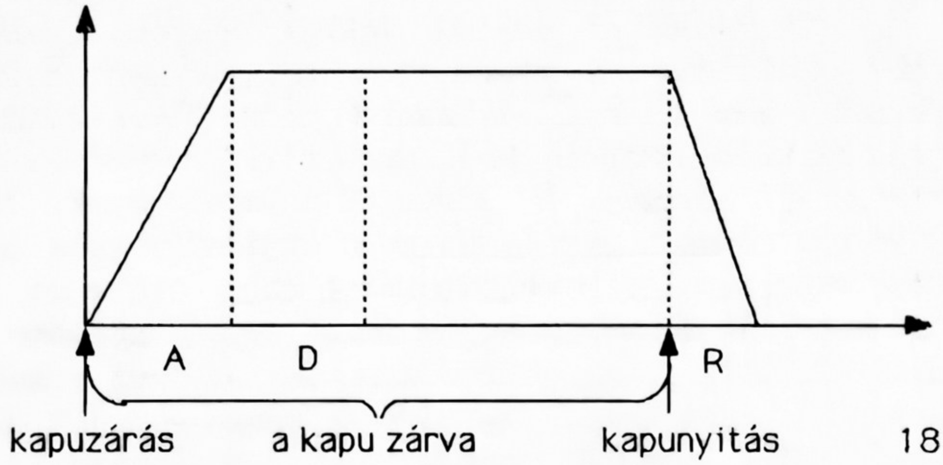
Nézzük meg, hogyan lehet a C-64-en az ADSR-t programozni! A négy paraméterhez a SID chip 4-4 bitet biztosít. Ezek elhelyezkedése:

Az első hang esetén:	S+5= 54277	felső 4 bitje	A
		alsó 4 bitje	D
	S+6= 54278	felső 4 bitje	S
		alsó 4 bitje	R

A második hang esetén hasonló megosztásban az S+12= 54284
és az S+13= 54285

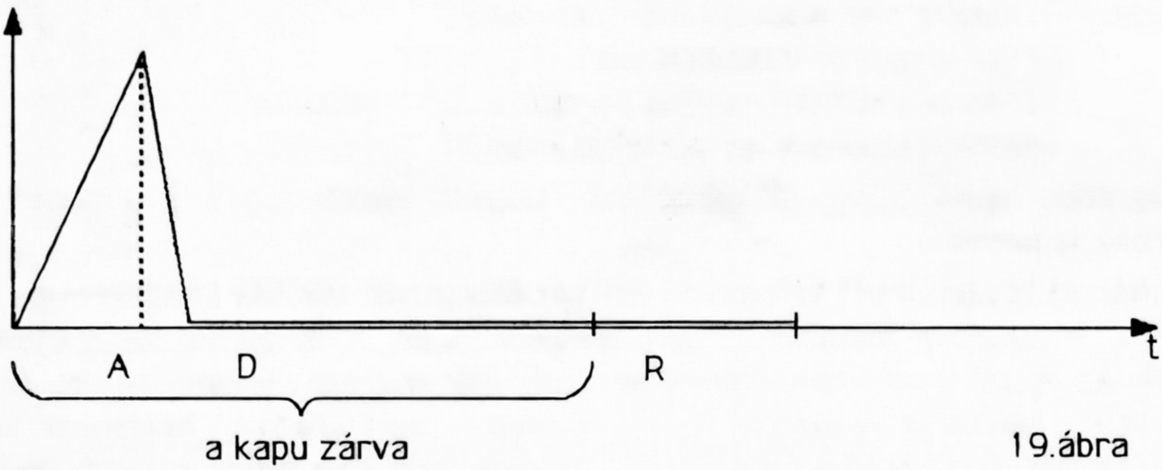
A harmadik hang esetén hasonló megosztásban az S+19= 54291
és az S+20= 54292

Így mind a négy paraméter 0-15 közötti egész értéket vehet fel. Az S=15 érték azt jelenti, hogy a kitartási szint maximális, azaz megegyezik a maximális hangerővel.



18. ábra

S=0-nál a kitartási szint 0, tehát ilyenkor az R értékét tetszőlegesen megválaszthatjuk, hiszen az elhalkulási szakaszban már úgysem fogunk semmit sem hallani.



19.ábra

A következő adatok - bár nincs igazán nagy jelentőségük - nagy vonalakban képet adnak az A, D, R értékekhez tartozó valós időtartamokról.

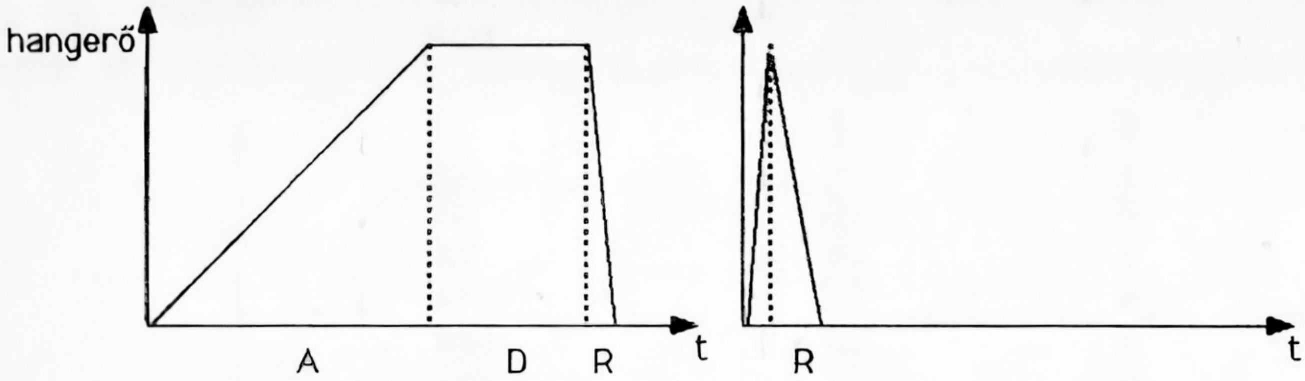
Érték	Felfutási idő ms-ban (A)	Lecsengési idő/elengedési idő ms-ban (D) (R)
0	2	6
1	8	24
2	16	48
3	24	72
4	38	114
5	56	168
6	68	204
7	80	240
8	100	300
9	250	750
10	500	1500
11	800	2400=2,4 s
12	1000=1 s	3000=3 s
13	3000=3 s	9000=9 s
14	5000=5 s	15000=15 s
15	8000=8 s	24000=24 s

Forrás: [1] és [13]

A táblázat alapján azt gondolhatnánk, hogy ha pl. A-nak 15-ös értéket választunk, akkor a felfutási idő 8 másodperc lesz. Ez azonban csak akkor van így, ha a kaput legalább nyolc másodpercig zárva tartjuk, ugyanis amint kinyitjuk - azaz a GATE-bit értéke 1-ről 0-ra változik - elkezdődik a hang elengedése. Erre mutat példát a következő program.

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,15
25 POKES+5,252    :REM 252=16*15+12,IGY A=15 D=12
30 POKES+6,244    :REM 244=16*15+4 ,IGY S=15 R= 4
35 POKES+4,17     :REM HAROMSZOG HULLAMFORMA ES KAPUZARAS
40 POKES+1,24:POKES,157:REM HANGMAGASSAG F#-4
45 POKES+4,32     :REM FURESZFOG HULLAMFORMA ES KAPUNYITAS
```

Hiába állítottuk be a 25-ös sorban a felfutási időt maximumra (A=15) és a lecsengési időt kb 3 másodpercnyire (D=12), olyan hamar nyitjuk ki a kaput, hogy mindebből szinte semmit sem hallunk. A 35-ös sor miatt e rész hullámformája háromszög, míg a kapu megnyitása után a megszólaló hang hullámformája fűrészfog (45-ös sor),hogy az előbbbitől megkülönböztethető legyen. Hagyjuk el a programból a 25-ös sort, és futtassuk így. Az eredmény ugyanaz lesz, mint az előbbi, pedig most A=0, D=0.



20.ábra a b

MFeladat 3.

Bővítsük az előbbi programot egy újabb sorral: 50 GOTO 35. Futtassuk a programot, majd a RUN/STOP billentyűvel állítsuk meg. Vonjuk le a tanulságot!

Feladat 4.

A következő programban különböző A, D, S, R értékeket próbálunk ki. A program futása előtt az AD és SR paraméterek értékeiből számítsuk ki az A, D, S és R értékeket, és rajzoljuk meg a hangerőgrafikonokat! A FOR I = 1 TO 5000: NEXT I üres ciklus kb. 5 mp-et vesz igénybe.

```
5 REM   A 2.HANGOT KULONBOZO A,D,S,R ERTEKEKKEL FOGJUK
6 REM   MEGSZOLALTATNI
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,5              :REM HANGEROSSEG 5
25 FORJ=1TO8               :REM 8 PAR ADATOT FOGUNK BEOLVASNI
30 READ AD,SR:PRINT AD;SR
35 POKES+12,AD:POKES+13,SR:REM AD=16*A+D  SR=16*S+R
40 POKES+11,17               :REM HAROMSZOGHULLAM ES
41 REM                            KAPUZARAS
45 POKES+8,17:POKES+7,103   :REM HANGMAGASSAG:C-4
50 FORI=1TO5000:NEXTI        :REM A HANG KITARTASA
55 POKES+11,32               :REM FURESZFOGHULLAM ES
56 REM                            KAPUNYITAS
60 FORI=1TO5000:NEXTI        :REM A HANG ELENGEDESERE
61 REM                            BIZTOSITOTT IDO
65 NEXTJ
70 DATA 240,0, 15,0, 170,0, 0,240
75 DATA 0,255, 1,27, 17,0, 188,77
```

MFeladat 5.

A következő program futtatásakor nem hallunk hangot. Keressük meg a hibát és vonjuk le a tanulságot!

```
10 S=54272
15 FORI=0TO23:POKES+I,0:NEXT
20 POKES+24,15
25 POKES+4,17
30 POKES+5,255:POKES+6,143
35 POKES+1,19:POKES,121
40 FORI=1TO10000:NEXTI
45 POKES+4,32
50 END
```

Feladat 6.

Módosítsuk a következő programban a 35-ös és az 50-es sorok üres ciklusainak idejét először a táblázat alapján, majd önállóan! Magyarázzuk meg a hallott jelenséget!

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,15
25 POKES+6,255
30 POKES+1,69:POKES,29:POKES+4,17
35 FORI=1TO 5:NEXT
40 POKES+4,16
45 POKES+1,28:POKES,49:POKES+4,17
50 FORJ=1TO 5:NEXT
55 POKES+4,16
60 GOTO30
```

35 FOR I= 1 TO	5	5	5	50	500	1000
50 FOR J= 1 TO	5	10	100	100	500	1000
megjegyzés:	nem halkul el	fokozatosan elhalkul	"helikopter (felszállás előtt)"	hol a magas, hol a mély, hol pedig mindkettő		

2.5 Szűrés és rezonancia

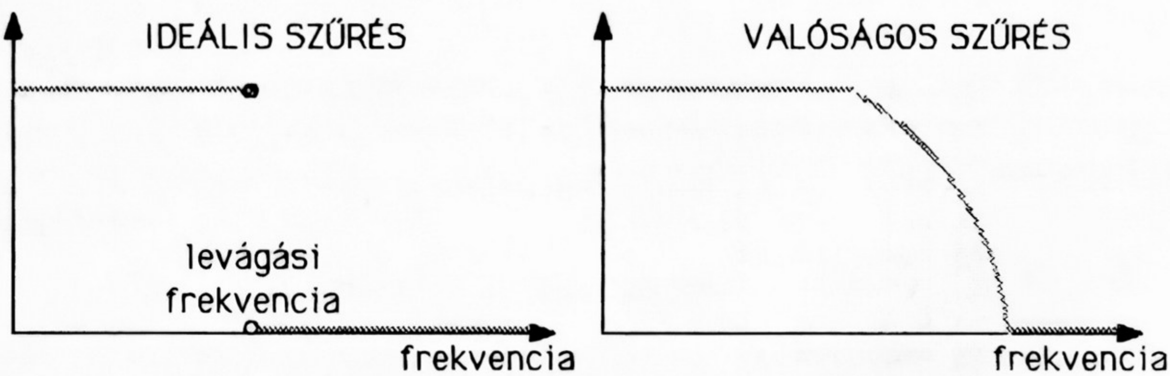
Egy hang hangszínét, hullámformáját, hangspektrumát szűréssel tovább lehet módosítani.

A szűrőknek több fajtája létezik, melyeket aszerint különböztetünk meg, hogy melyik frekvenciasávot szűrik ki:

- mélyáteresztő, aluláteresztő szűrők
- magas- vagy felüláteresztő szűrők
- sáváteresztő szűrők
- sávvisszatartó-, sávzáró szűrők.

a) Mélyáteresztő, aluláteresztő szűrők

Mint ahogy a mélyáteresztő név jól kifejezi -, lényegében csak a szűrés előre megadott levágási frekvenciája alatti, mélyebb hangok jutnak át a szűrőn.

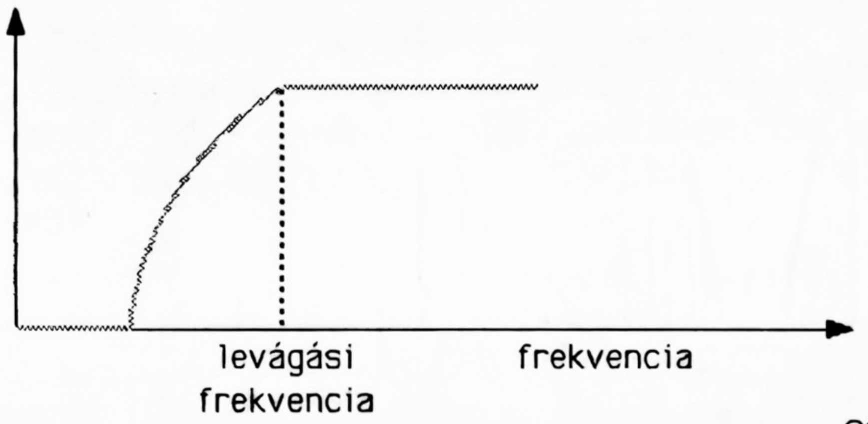


21.ábra a b

E szűrő segítségével tiszta szinuszos jelet is létrehozhatunk. Ugyanis, ha pl. egy "normál a" hangot állítunk elő (melynek 440 Hz a frekvenciája, így legmélyebb felharmonikusa 2·440 Hz=880 Hz-es) akkor legyen bármilyen hullámformája is, a szűrés levágási frekvenciáját 440 Hz-nél valamivel nagyobb értékre állítva mélyáteresztő szűrővel szinuszos "normál a"-t kapunk.

b) Magas- vagy felüláteresztő szűrők

Felüláteresztő szűrés esetén éppen a fordítottja történik. Most nem a levágási frekvencia alatti hangok jutnak túl a szűrőn, hanem az e felettiek.

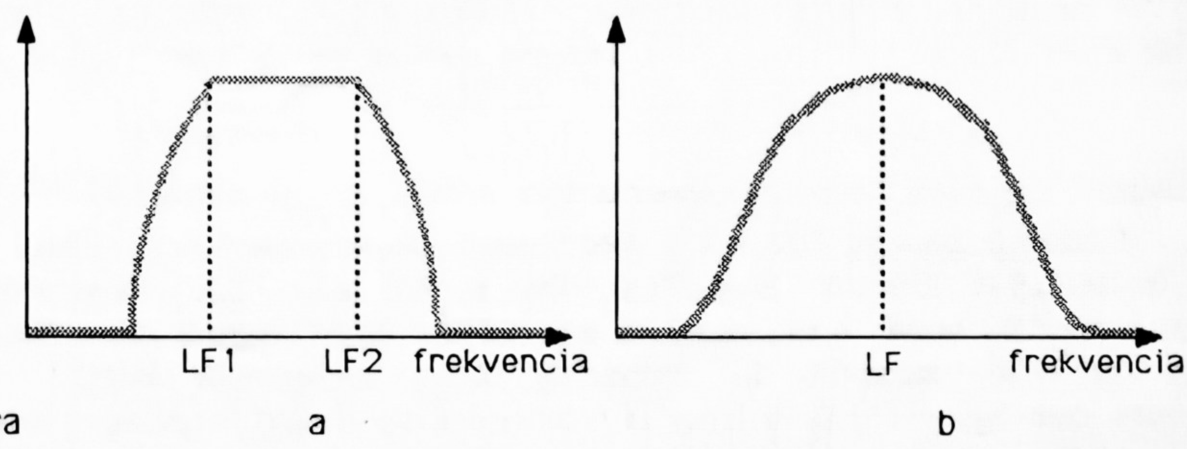


22.ábra

Ezért, ha pl. a levágási frekvencia kisebb az alaphang frekvenciájánál, akkor a felüláteresztő szűrő nem változtat semmit. Ha viszont az alaphang frekvenciájának kétszeresénél egy kicsivel kisebb, akkor felüláteresztő szűrő esetén az alaphangot nem, csak a felhangjait lehet hallani.

c) Sáváteresztő szűrők

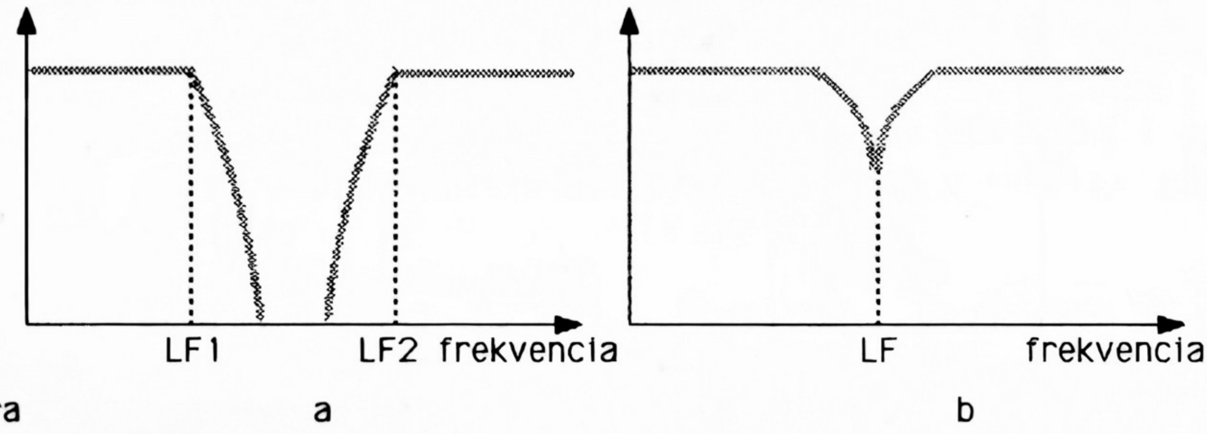
Sáváteresztő szűréskor általában két frekvenciaparaméterre van szükség. (LF1, LF2) Ezek határozzák meg azt a frekvenciasávot, amelyen kívül eső hangok nem juthatnak át a szűrőn. (Lásd 23/a ábra) A C-64, azaz a SID chip egy újabb hiányossága, hogy ebben az esetben is csak egy levágási frekvenciát használ, és az e körüli frekvenciasávba tartozó hangokat engedi át. (23/b ábra)



23.ábra a b

d) Sávvisszatartó-, sávzáró szűrők

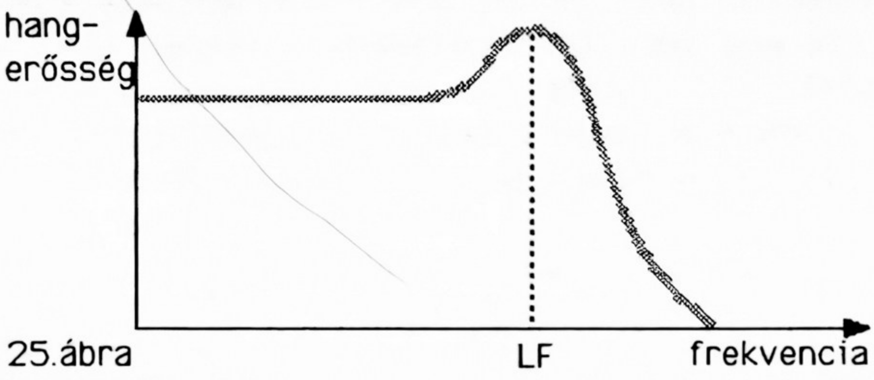
A sávvisszatartó szűrés esetén is két paraméter szükséges ahhoz, hogy a szűrési frekvenciasávot meghatározzuk. (24/a ábra) A C-64-en az alul- és felüláteresztő szűrők együttes használatával érhetjük el, hogy a levágási frekvencia közvetlen közelében lévő hangokat kiszűrjük. (24/b ábra)



24.ábra

A figyelmes Olvasó ellentmondást vehet észre a 24/b és az előző ábrák között. A 21/b és a 22. ábra valóban pontatlan, mert a C-64-en a magas- és mélyáteresztő szűrés nem pontosan a levágási frekvenciánál kezdődik - éppen azért, hogy a sávvisszatartó szűrés is megvalósítható legyen.

A szűrés kapcsán érdemes megismerkednünk egy másik fontos jelenséggel, a rezonanciával is. Rezonancia esetében - a szűréssel ellentétben - a megadott levágási frekvencia környékén a hangok hangereje megnő, így ezek nagyobb hangsúlyt kapnak. A következő ábra aluláteresztő szűrés mellett mutatja be a rezonanciát.



25.ábra

Nézzük meg, hogy C-64-en hogyan tudjuk programozni a szűrést és a rezonanciát!

1./ Először is be kell állítanunk, hogy a három kezelhető hang közül melyiket, vagy melyikeket akarjuk megszűrni. Ezt a SID chip 23. regiszterén tudjuk meghatározni. Ha ennek a regiszternek a 0. bitje, 1. bitje, vagy 2. bitje magas, akkor rendre az első, második, ill. harmadik hang kimenőjele kerül a szűrőbe. Természetesen egyszerre több hang is szűrhető a megfelelő bitek együttes magasra állításával. Ha az említett bitek valamelyike alacsony, azaz 0, akkor a neki megfelelő hangot nem szűrjük meg. Az audio bemeneten* megjelenő jel szűrését ugyanezen regiszter 3. bitjének magasra állításával érhetjük el.

---

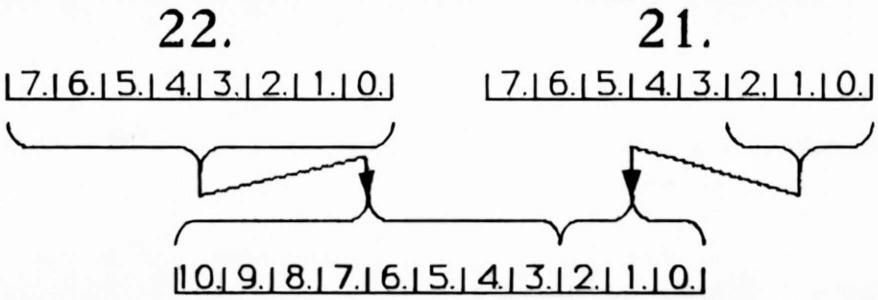
* Az audio bemenettel kapcsolatos tudnivalókat a későbbiekben tárgyaljuk részletesen.

2./Ezután ki kell választanunk valamelyik szűrésfajtát. Ezt a SID chip 24. regiszterén tudjuk beállítani, mégpedig:

ha ennek a regiszternek a 4. bitje magas, akkor a szűrő aluláteresztő,
ha az 5. bitje magas, akkor sáváteresztő,
ha a 6. bitje magas, akkor felüláteresztő módban működik.

Mint már volt róla szó, a szűrés módjai keverhetők is.

3./ Meg kell határoznunk a szűrés frekvenciáját, a levágási frekvenciát. Erre a célra a SID chip 21. regiszterének 0-2 bitjei, és a teljes 22. regiszter szolgál. (Ez utóbbiak a magasabb bitek.)



26. ábra

Az így kapott 11 jegyű bináris szám egyenesen arányosan határozza meg a levágási frekvenciát hozzávetőlegesen 30 Hz és 12 kHz között.

4./ Végül beállíthatjuk a rezonancia nagyságát is. Ezt a SID chip 23. regiszterének 4.-7. bitjei által meghatározott 4 jegyű bináris szám (0-15) szabályozza, mégpedig lineárisan. Természetesen a 15-ös értékhez tartozik a maximális rezonancia, míg a 0 érték esetén egyáltalán nincs. A gyakorlatban érdemes ezt a lépést az elsővel egyidejűleg végezni, hiszen mindkét esetben a SID 23. regiszterét módosítjuk.

MFeladat 7.

Adjuk meg a levágási frekvenciát az őt meghatározó 11 jegyű bináris szám függvényeként, és fordítva. Ennek alapján számítsuk ki, hogy a 440 Hz-es levágási frekvenciához hozzávetőlegesen milyen értékeket kell a SID 21. és 22. regiszterébe írnunk.

MFeladat 8.

Határozzuk meg az egyes programsorok jelentését a következő programban!

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,31
25 POKES+12,9:POKES+13,172
30 POKES+10,8
35 POKES+23,2
40 FORJ=0TO255 STEP 5
45 POKES+22,J
50 POKES+11,65
55 POKES+8,14:POKES+7,162
60 FORI=1TO500:NEXTI
65 POKES+11,64
70 NEXTJ
```

Miután már a program minden utasításának jelentését ismerjük, kísérletezzünk az egyes paraméterek változtatásával! Érdemes a következő módosításokat elvégezni:

a/ AD = 9 SR = 0 a 25-ös sorba
b/ AD = 0 SR = 255 (25)
c/ FOR I = 1 TO 100 (60)
d/ FOR I = 1 TO 50 (60)
e/ Rezonancia maximálisra állítása POKE S + 23, 2 + 15 ·16 (35)
f/ a különböző szűrésfajták kipróbálása (20)
g/ különböző hullámformák szűrése (50;65)
h/ a megszólaló hang magasságát, frekvenciáját változtatni

M Feladat 9.

Hogyan lehet a rezonanciát pl. 10-es értéknek megfelelőre állítani, hogy közben semmi más értéket ne változtassunk?

2.6 Szinkronizáció és körmodulálás

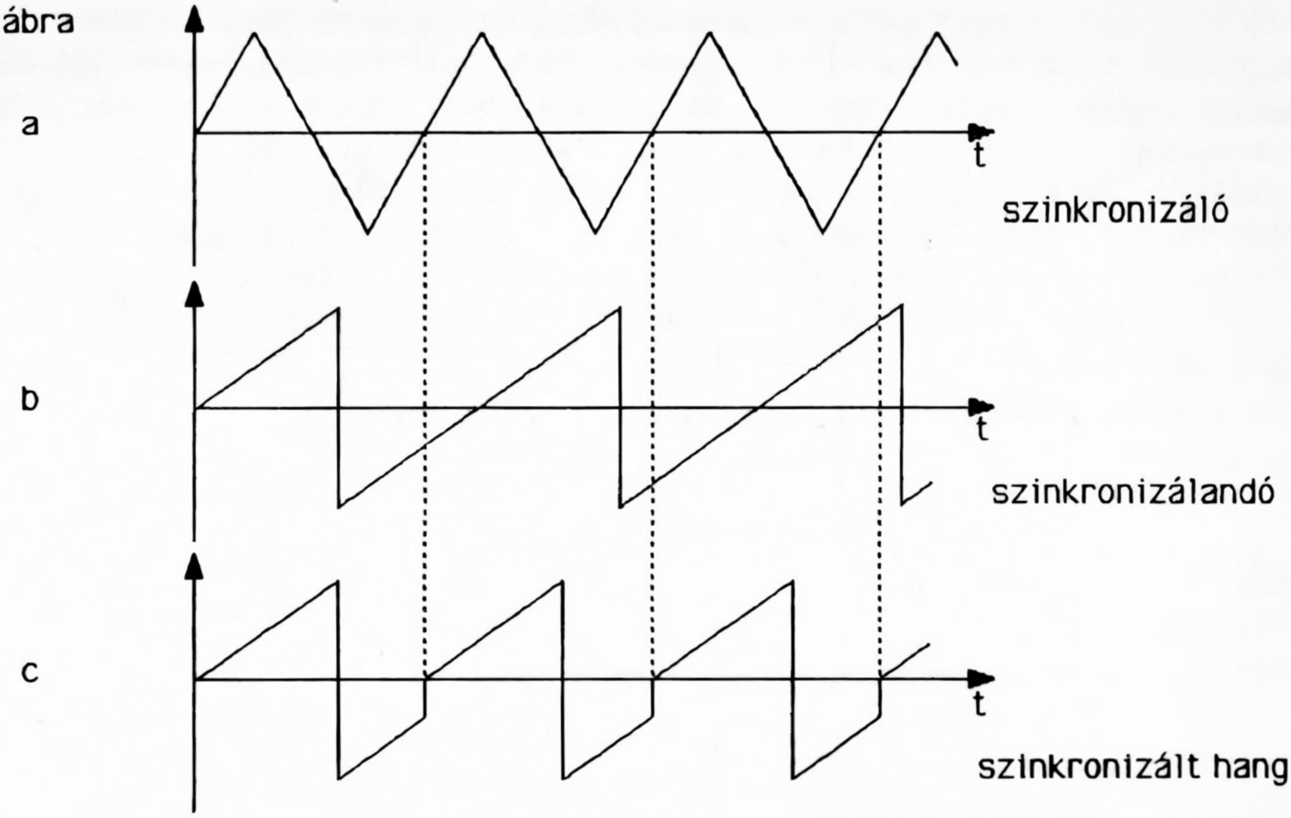
Szinkronizációhoz és körmoduláláshoz is 2 hanggenerátor szükséges. C-64 használatakor a következő táblázatból olvashatjuk le, hogy melyik hanggal melyik hangot szinkronizálhatjuk, ill. modulálhatjuk.

szinkronizálandó/modulálandó hang	1	2	3
szinkronizáló/moduláló hang	3	1	2

a) Szinkronizálás

Szinkronizáláskor a szinkronizálni kívánt hang kimenő jele függ a szinkronizáló hang frekvenciájától is.

27.ábra



Mint az ábrából is látható, a szinkronizálás azt jelenti, valahányszor a szinkronizáló hang új periódust kezd, ugyanezt teszi a szinkronizált hang is, függetlenül attól, hogy az előző periódust befejezte-e már, vagy sem. Így a szinkronizált hang frekvenciája és hullámformája is módosul. A C-64-nél a szinkronizáció csak akkor következik be, ha a szinkronizáló hang frekvenciája kisebb a szinkronizálandóénál, és természetesen, ha ez a frekvencia nem nulla. A szinkronizációt a megfelelő kontroll-regiszter 1. bitjének magasra állításával érhetjük el.

b) Körmoduláció

Körmodulációnál a SID chip a két hanggenerátor, oszcillátor jeléből egy olyan új jelet hoz létre, mely tartalmazza a két alaphang, valamint felhangjaik frekvenciáinak összegét és különbségét is. Pl. ha egy 400 Hz frekvenciájú szinuszrezgést modulálnak egy 100 Hz frekvenciájú szinuszjellel, akkor a modulált hang egy 400+100=500 Hz, és egy 400-100=300 Hz frekvenciájú hang összegeként állítható elő. (Így, ha a moduláló hang frekvenciája 0, akkor a modulálásnak nincs semmiféle hatása.) Miután a modulált hang nem alaphangból és annak felharmonikusaiból áll, ezért a modulált jel igen komplex is lehet.

Általános esetben, a körmodulációval létrehozott jel teljes mértékben a két alaphang frekvenciájától és a két hang színképétől (hullámforma, szűrés, rezonancia, stb.) függ. A C-64-nél azonban fontos megkötés, hogy a modulálandó hang számára háromszög hullámformát kell választanunk, csak ekkor történik modulálás.

Körmodulálást a megfelelő kontroll-regiszter 2. bitjének magasra állításával hozhatunk létre. Mind a szinkronizációnál, mind a körmodulációnál a szinkronizáló, ill. a moduláló hang frekvenciájának nullától különbözőnek kell lennie. Ha ennek a hangnak egyéb paramétereit is beállítottuk már, akkor előfordulhat, hogy a szinkronizáló, ill. moduláló hang is megszólal, amikor mi csak a szinkronizált, ill. modulált hangra lennénk kíváncsiak. Abban az esetben, ha ezt a hangot a 3. hanggenerátor állítja elő, akkor a 24. regiszter 7. bitjének magasra állításával elérhetjük e hang leválasztását az audio kimenetről. A másik két hanggenerátornál úgy érhetjük el célunkat, hogy a szinkronizáló, ill. moduláló hang számára nem választunk hullámformát.

A következő program egy körmodulált hangsort mutat be.

```
5 REM  AZ 1.HANGOT FOGJUK MODULALNI
6 :
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,15          :REM  HANGEROSSEG MAX.
25 POKES+6,255          :REM  S=15 R=15 AZ 1.HANG ESETEN
30 POKES+20,255         :REM  S=15 R=15 A  3.HANG ESETEN
35 POKES+15,17:POKES+14,100:REM A 3.HANG HANGMAGASSAGA
40 POKES+18,33          :REM A 3.HANG HULLAMFORMAJA
41 REM                       FURESZFOG;KAPUZARAS
45 FORJ=1TO12
50 READ FB,AB
55 POKES+1,FB:POKES,AB:REM AZ 1.HANG HANGMAGASSAGA
60 POKES+4,21           :REM 21=16+4+1 AZAZ HAROMSZOG
61 REM                       HULLAMFORMA,KORMODULALAS ES
62 REM                       KAPUZARAS
65 FORI=1TO700 :NEXT
70 NEXTJ
75 POKES+24,0           :REM HANGEROSSEG 0
80 DATA 17,103, 18,112, 19,137, 20,178
85 DATA 21,237, 23, 59, 24,157, 26, 20
90 DATA 27,160, 29, 69, 31,  3, 32,219
```

A program lefuttatása után végezzük el a következő módosításokat:

- 20 POKE S + 24, 15 + 128. Ennek hatása: a 3. hang kimenőjele megszűnik.
- A 40-es sorból töröljük a POKE S + 18, 33 utasítást, majd a 30-asból a POKE S + 20, 255 utasítást. A program így is működni fog, tehát jelen esetben valóban elegendő a moduláló hangnak csak a frekvenciáját beállítani.
- Módosítsuk a moduláló hang frekvenciáját a 35-ös sorban:

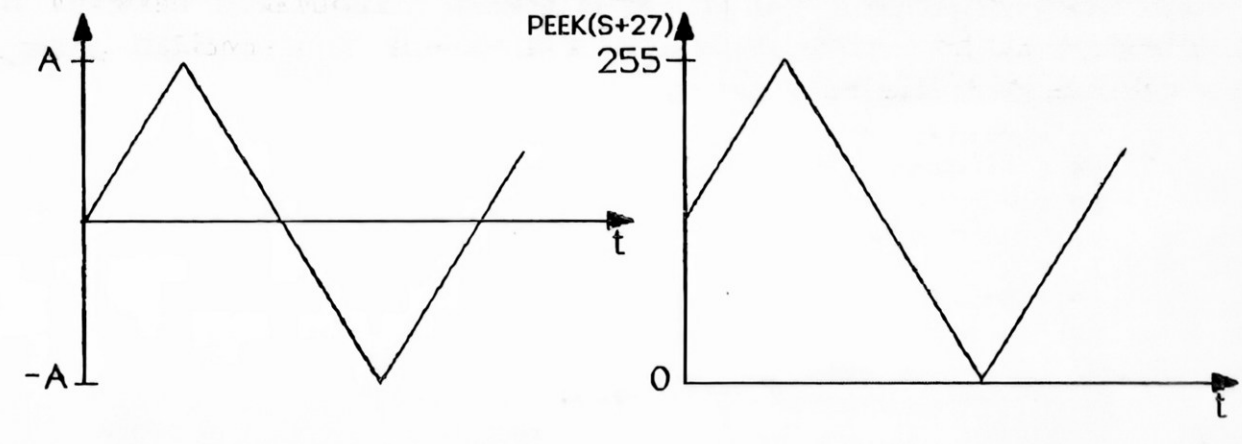
  Pl.: (47, 100)
  (98, 100) csökkenő hangsor érzete
  (0, 0) mintha nem lenne moduláció
  (19, 137) a 3. megszólaló hangon nem lesz igazán érzékelhető a modulálás.
- állítsuk át a 60-as sorban a modulálandó hang hullámformáját pl. fűrészfogra: POKE S + 4, 33 + 4
  Jelen esetben sincs a körmodulációnak hatása.
- Módosítsuk a körmodulációt szinkronizálásra
  Pl. POKE S + 4, 35 és legyen a szinkronizáló hang frekvenciája kisebb, pl. (0, 20),

(0, 180), (1, 186), (2, 180), (3, 180), (10, 180); illetve nagyobb, mint a szinkronizált hang frekvenciája, pl. (50, 0).

A SID írható regisztereivel kapcsolatban már csak egy dolgot kell tudnunk. A kontroll regiszterek 3. bitjének magasra állításával a megfelelő oszcillátort lezárjuk, s addig nem tudjuk használni, míg azt a bitet ki nem nullázzuk. Ezt akkor lehet jól használni, ha egy többszólamú darabban, – kiváltképp, ha programját BASIC-ben írtuk – kíváncsiak vagyunk egyes szólamokra. Így ugyanis bármely szólam kikapcsolható.

2.7 A SID csak olvasható regiszterei

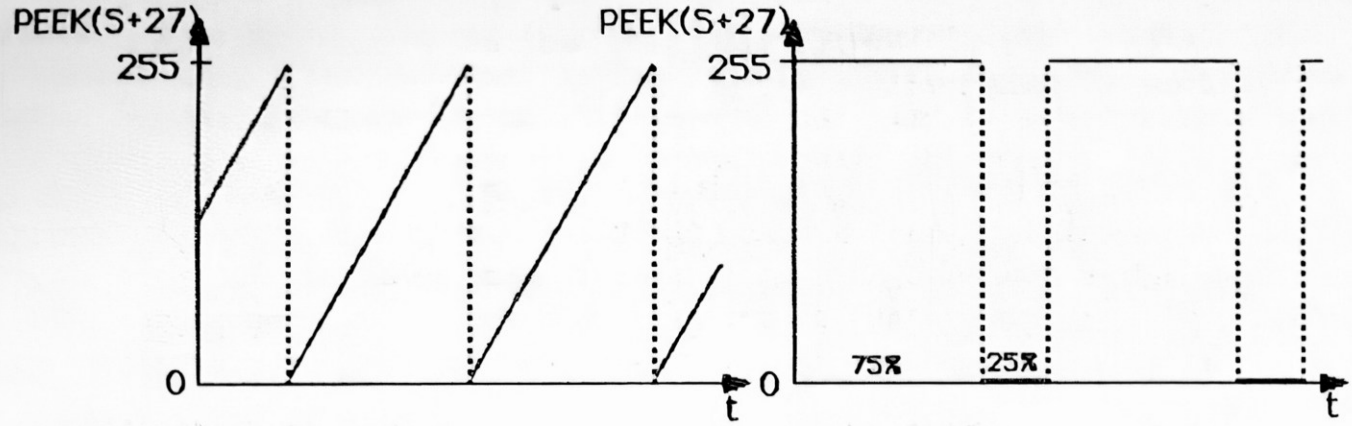
a) Az S + 27 = 54299. byte, azaz a SID chip 27. regiszterének értéke minden pillanatban a 3. oszcillátor keltette hang rezgésétől függ. A kettő közti kapcsolatot mutatja a következő ábra, ha a 3. hang hullámformája háromszög hullám.



28.ábra a b

Ilyenkor tehát a 27. regiszter értéke fokozatosan halad 0-255-ig, azután vissza, majd ismét 0-255-ig, stb. A változás sebessége természetesen egyenesen arányos a rezgés frekvenciájával. A 3. hang frekvenciájának növelésével a 27. regiszter ertéke is gyorsabban változik.

A következő ábrákon a fűrészfog és 25%-os négyszög hullámra láthatjuk a SID 27. regiszter értékének alakulását. Az utóbbi esetben az ábrának pont a fordítottját várnánk. Az eltérés magyarázatát többfelé kereshetjük, azonban ettől eltekintünk. A Kedves Olvasó azért elgondolkozhat a problémán.



29.ábra a b

Fehérzaj hullámforma esetén a 27. regiszter értéke lényegében egyenletes eloszlású a 0-255 közötti egészeken, s így véletlenszám-generálásra is használhatjuk. A gyakorlatban erre ritkán kerül sor, hiszen az RND függvény kezelése jóval egyszerűbb.

A következő program a SID 27. regiszterének működését illusztrálja. A 3. hang hullámformájának háromszög hullámot választottunk, frekvenciáját pedig kicsinek, hogy a változás jól érzékelhető legyen.

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,15
25 POKES+6,255:POKES+19,9
30 POKES+17,4
35 POKES+4,33:POKES+18,16
40 POKES+14,2          :REM A 3.HANG FREKVENCIAJA
45 FORJ=1TO1500
50 POKES+1,PEEK(S+27)  :REM AZ 1.HANG FREKVENCIAJA A
51 REM                      A SID 27.REGISZTERENEK
52 REM                      ERTEKEVEL EGYUTT VALTOZIK
55 NEXT
60 POKES+24,0
```

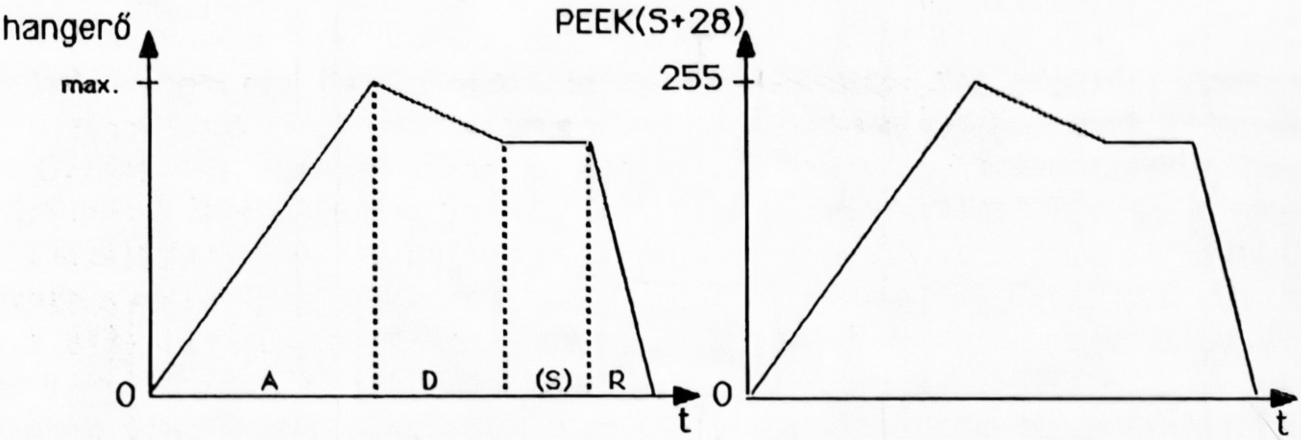
Feladat 10.

a./ Változtassuk a 3. hang hullámformáját a 35-ös sorban.
b./ Változtassuk a 3. hang frekvenciáját a 40-es sorban.
c./ Változtassuk a cikluson belül a 3. hang frekvenciáját is a 27. regiszter segítségével.
d./ Moduláljuk az 1. hangot a 3.-kal.
e./ Állítsunk elő 10 és 20 között véletlen egész számokat a fenti program segítségével. Folytassuk a kísérletezést önállóan!

b) Az S + 28 = 54300. byte, azaz a SID 28. regisztere nagyon hasonlít az előzőhöz. A különbség mindössze az, hogy most a 3. hang burkológörbéjének (ADSR) megfelelően változik a regiszter értéke. A következő program jól illusztrálja a regiszter értékének változását. A 30-as sorban állítjuk be a 3. hang számára a

burkológörbét (A=11, D=13, S=10, R=10). Ennek megfelelően alakul az 1. hang frekvenciája is az 50., ill. 70-es sor hatására. Futtassuk le a programot!

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 POKES+24,15
25 POKES+6,255
30 POKES+19,189:POKES+20,170
35 POKES+17,4
40 POKES+4,33:POKES+18,17
45 FORJ=1TO500
50 POKES+1,PEEK(S+28)
55 NEXT
60 POKES+18,16
65 FORJ=1TO200
70 POKES+1,PEEK(S+28)
75 NEXT
80 POKES+24,0
```



30.ábra a b

Figyeljük meg, hogy ha a hangerő már lecsengett a közbülső szintre, de a KAPU-bit értéke még 1, akkor a hangerővel együtt a 28. regiszter értéke is változatlan marad a KAPU-bit nullára állításáig. (60-as sor)

Feladat 11.

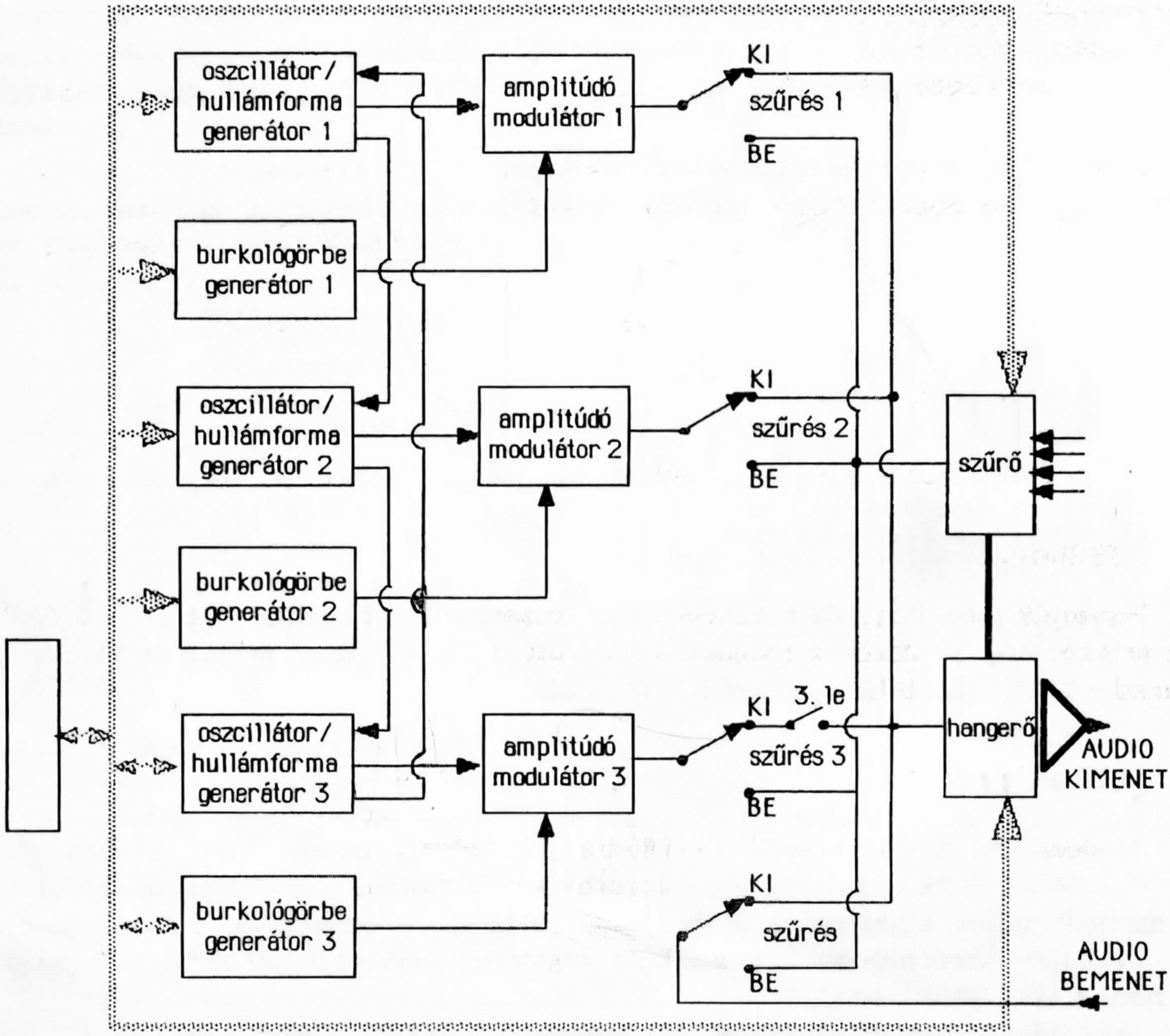
a./ Módosítsuk a 40-es sort! (S+18)-ba 17 helyett írjunk 16-ot. Futtassuk a programot! Noha a 3. hang burkológörbéje a 4. fázisban aktivizálódna (közbülső szintről teljes elhalkulás), a program futásakor mégsem hallunk semmit. A tanulságot máris levonhatjuk: a SID 28. regisztere csak akkor aktivizálódik, ha a 3. hang KAPU-bitje 1-es.

b./ Állítsuk vissza "hallhatóvá" a programot, majd változtassuk a 3. hang burkológörbéjét (30-as sor).

c) Már csak a SID 25. és 26. regiszterének ismertetése lenne hátra, de mivel ezek a regiszterek a zenélésben nem használatosak, így ettől eltekintünk. Röviden csak annyit, hogy az előbbi a SID chip POTX, míg az utóbbi a SID chip POTY bemenetéhez kapcsolt potenciométerek időbeli változását mutatja. Az érdeklődők az [1] 6.6 és 8.4 fejezetében, ill. a [13] 235-236. oldalán részletesebben megismerkedhetnek ezen regiszterek használatával.

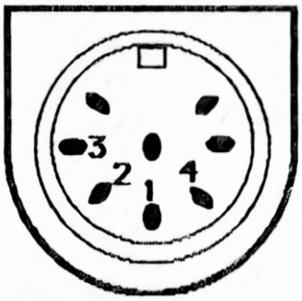
2.8 A SID chip blokkdiagramja

A fejezet befejező részében a SID chip blokkdiagramja alapján két kiegészítő megjegyzést fűzünk az eddig elmondottakhoz.



a) Bármely hang dinamikáját (ADSR) egy amplitúdó-modulátor, és egy ezt kontrolláló burkológörbe-generátor szabályozza. E kettő együttes működésének köszönhető, hogy az oszcillátort elhagyó hang csak bizonyos megszakításokkal jut keresztül az amplitúdó modulátoron.

b) A második megjegyzés az audio bemenettel kapcsolatos. Nem történt elírás: a C-64 valóban rendelkezik audio bemenettel. Ez lényegében azt jelenti, hogy bármilyen hangforrást (mikrofont, elektromos gitárt, másik C-64-et, lemezjátszót, magnót, stb.) rákötheтünk gépünkre. Hogyan és hol? A nulladik fejezet harmadik pontjában már említettem, hogy hová kell az átjátszózsinórt csatlakoztatni, ha számítógépünket erősítővel akarjuk összekötni. Vessünk egy pillantást erre a helyre! Bár ide egy nyolclábú "zsinór-szörnyet" is csatlakoztathatnánk, a nyolc láb közül csak négy érdekes számunkra.



1 föld (GND = ground)
2 audio bemenet (audio input, external input)
3 audio kimenet (audio output)
4 video kimenet (video output)

Ennek alapján egy forrasztópáka és egy átjátszózsinór segítségével néhány perc alatt létrehozhatjuk az összeköttetést a külső hangforrás és a C-64 között. (**Vigyázat! Egy rövidzárlat tönkreteheti a gépet!**) A blokkdiagramból kiderül, hogy a bemenő jelet is meg tudjuk szűrni. Így akár az is megvalósítható, hogy egy program – a szükséges paraméterek beállítása után – változtatja a levágási frekvenciát, miközben a számítógéphez kapcsolt mikrofonba énekelünk. Ugyanúgy kell eljárnunk, mint a SID által előállított hangoknál: kiválasztjuk a szűrés típusát, beállítjuk a levágási frekvenciát, és a szűrőre küldjük az audio bemeneten megjelenő jelet. A rezonancia értékét is megadhatjuk 0 és 15 között. Az audio bemenetről jövő jel hangerejét is a SID 24. regisztere határozza meg.

Az igazsághoz tartozik, hogy pl. mikrofon használata esetén megfelelő hanghatás csak úgy érhető el, ha a C-64 és a külső hangforrás közé egy erősítőt illesztünk.

A következő program egy C-64-gyel összekötött hangforrás jelének szűrésére mutat példát:

```
10 S=54272
15 FOR I=0 TO 24:POKES+I,0:NEXT
20 POKES+24,15+16   :REM HANGERO MAX. ES ALULATERESZTO SZURES
25 POKES+23,16      :REM AZ AUDIO BEMENETROL JOVO HANGOT
26 REM                   SZURORE KULDJUK
30 FOR LF=0 TO 2047:REM 2047=2↑11-1 EZ A SZURES LEVAGASI
31 REM                   FREKVENCIAJAT MEGHATAROZO
32 REM                   PARAMETER MAX.ERTEKE
35 PRINT LF
40 POKES+22,INT(LF/8)     :REM A LEVAGASI FREKVENCIA
45 POKES+21,LF-8*INT(LF/8):REM      BEALLITASA
50 NEXT
```

Ezzel a SID chip lehetőségeinek ismertetését befejeztük, s a következő táblázatban már csak összefoglaljuk a regiszterek tartalmát.

		7	6	5	4	3	2	1	0
	S	P7	P6	P5	P4	P3	P2	P1	P0
	S+1	P15	P14	P13	P12	P11	P10	P9	P8
	S+2	I7	I6	I5	I4	I3	I2	I1	I0
1	S+3	-	-	-	-	I11	I10	I9	I8
	S+4	ZAJ	■	⊥	◤	X	●	SZ	KAP
	S+5	A3	A2	A1	A0	D3	D2	D1	D0
	S+6	S3	S2	S1	S0	R3	R2	R1	R0
	S+7	P7	P6	P5	P4	P3	P2	P1	P0
	S+8	P15	P14	P13	P12	P11	P10	P9	P8
	S+9	I7	I6	I5	I4	I3	I2	I1	I0
2	S+10	-	-	-	-	I11	I10	I9	I8
	S+11	ZAJ	■	⊥	◤	X	●	SZ	KAP
	S+12	A3	A2	A1	A0	D3	D2	D1	D0
	S+13	S3	S2	S1	S0	R3	R2	R1	R0
	S+14	P7	P6	P5	P4	P3	P2	P1	P0
	S+15	P15	P14	P13	P12	P11	P10	P9	P8
	S+16	I7	I6	I5	I4	I3	I2	I1	I0
3	S+17	-	-	-	-	I11	I10	I9	I8
	S+18	ZAJ	■	⊥	◤	X	●	SZ	KAP
	S+19	A3	A2	A1	A0	D3	D2	D1	D0
	S+20	S3	S2	S1	S0	R3	R2	R1	R0
	S+21	-	-	-	-	-	L2	L1	L0
	S+22	L10	L9	L8	L7	L6	L5	L4	L3
	S+23	RE3	RE2	RE1	RE0	AUD	3SZ	2SZ	1SZ
	S+24	3LE	FEL	SAV	ALU	HE3	HE2	HE1	HE0
	S+25	POTX							
	S+26	POTY							
	S+27	A 3.HANG REZGESE							
	S+28	A 3.HANG BURKOLOGORBEJE							

A táblázatban a következő jelöléseket használtuk:

P	a megfelelő hang frekvenciáját meghatározó paraméter
I	négyszöghullám esetén az impulzusszélesség
■	négyszöghullám
⊥	fűrészfoghullám
◣	háromszöghullám
✕	a hang törése
●	körmoduláció
SZ	szinkronizáció
KAP	KAPU-bit (GATE-bit)
A	felfutási idő (attack)
D	lecsengési idő (decay)
S	közbülső szint (sustain)
R	a hang elengedése (release)
L	levágási frekvencia szűrés esetén
RE	rezonancia
AUD	az audio bemeneten megjelenő hangot szűrjük-e
3SZ;2SZ;1SZ	a 3., a 2., ill. az 1. hangot szűrjük-e
3LE	a 3. hang leválasztása az audiokimenetről
FEL, SAV, ALU	a szűrés fajtái
HE	hangerősség

M Feladat 12.

Készítsünk metronómot!

3. fejezet

Dallamok programozása

A dallamok programozásának két alapkérdése:

1. hogyan adjuk meg a hangok magasságát?
2. hogyan adjuk meg a hangok megszólalásának időtartamát?

Mivel a SID chip lehetővé teszi a hang burkológörbéjének programozását így a második kérdés két részre bontható:
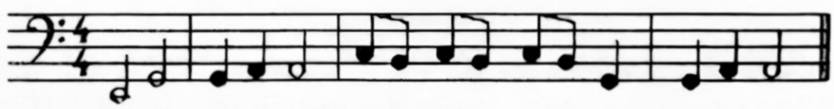
2a. hogyan adjuk meg a hang megszólalása és a kapu nyitása közötti idő nagyságát, vagyis a kitartási időt?

2b. hogyan adjuk meg azt az időt, ameddig a hang elengedése (R) tarthat?

E fejezetben ezekre a kérdésekre keresünk választ. A példaprogramok különböző megoldási lehetőségeket mutatnak be.

1. példa

A program a Black Sabbath együttes Vasember c. számának fő dallamát játssza el háromszor:



```
10 REM    VASEMBER
11 REM /CSIKOS ZS./
15 S=54272:V=0
20 FORI=0TO24:POKES+I,0:NEXT
25 POKES+24,15
30 POKES+5,8:POKES+6,169
35 FORJ=1TO15
40 READ P,K
45 IFP=0THENFORI=1TOK:NEXT:GOTO40
50 F%=P/256:A%=P-256*F%
55 POKES+1,F%:POKES,A%
60 POKES+4,33
65 FORI=1TOK:NEXTI
70 POKES+4,32
75 FORI=1TO10:NEXTI
80 NEXTJ
85 V=V+1:IFV=3 THEN END
90 RESTORE:GOTO35
91 DATA 1403,500, 1669,500, 1669,100
92 DATA    0,100, 1873,200, 1873,500
93 DATA 2228,80  , 2103,80
94 DATA 2228,80  , 2103,80
95 DATA 2228,80  , 2103,80
96 DATA 1669,120,    0,100, 1669,170
97 DATA    0,40  , 1873,190, 0,10
98 DATA 1873,500
```

A programban a következő módszert használjuk:

1. A hang magasságát a frekvenciája által meghatározott P paraméterrel adjuk meg DATA sorokban.

2a. A hang kitartási idejét jelentő számot szintén DATA sorokban adjuk meg.

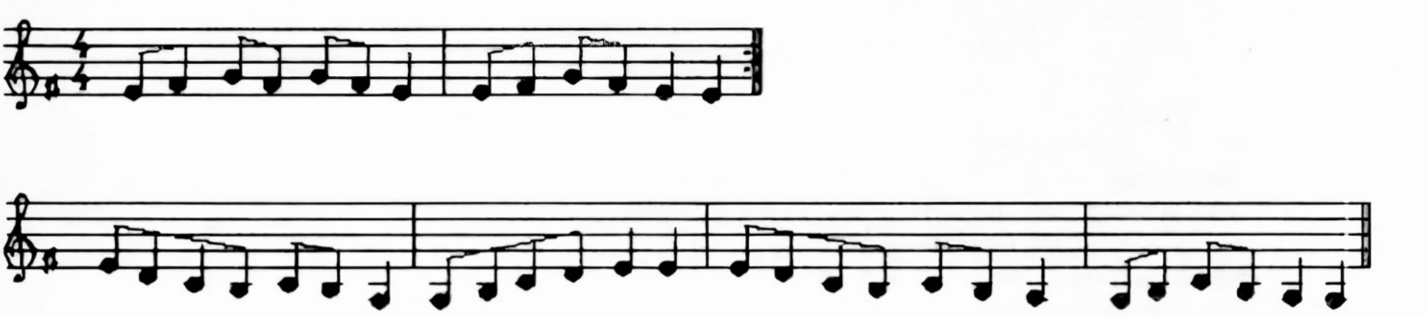
2b. Az elengedési időt a 75. sor üres ciklusa biztosítja, ez minden esetben ugyanannyi. Ehhez az időtartamhoz hozzáadódik a P paraméter feldolgozásához (50-es,55-ös sor) szükséges idő.

Érdemes megfigyelni a 45-ös sor szerepét: ez a sor kezeli a szünetet. Ha a beolvasott paraméter értéke 0, akkor a K értéknek megfelelő hosszúságú üres ciklust hajtja végre a program, és csak ezután következik az újabb hanghoz tartozó P és K érték beolvasása. A "szünet" persze nem feltétlenül jelenti azt, hogy ilyenkor semmit sem hallunk, hiszen az előző hang eközben is szólhat, ha teljes elengedése (R) még nem ért véget.

A 85-ös sorban számoljuk az ismétléseket. Ha a harmadik ismétlés is elhangzott (V=3), akkor befejeződik a program futása, ellenkező esetben a RESTORE hatására a következő READ utasítás ismét az első DATA sorból (91-es sor) fog olvasni.

2. példa

Ez a program addig játssza újra és újra a közismert "Még azt mondják nem illik a tánc a magyarnak..." kezdetű népdalt, amíg futását a RUN/STOP billentyűvel meg nem szakítjuk.



```
10 REM MEG AZT MONDJAK ...
11 REM /CSIKOS ZS./
15 S=54272
20 FORI=0TO24:POKE S+I,0:NEXT
25 POKE S+24,15
30 POKE S+5,9
35 READ F,A,K
40 IF F<0 THEN RESTORE:GOTO 35
45 POKE S+1,F:POKE S,A
50 POKE S+4,33
55 FOR I=1 TO K:NEXT
60 POKE S+4,32
65 GOTO 35
100 DATA 21,31,125, 23,181,125
101 DATA 25,30,125, 23,181,125
102 DATA 25,30,125, 23,181,125
103 DATA 21,31,250
104 DATA 21,31,125, 23,181,125
105 DATA 25,30,125, 23,181,125
106 DATA 21,31,250, 21,31,250
107 DATA 21,31,125, 23,181,125
108 DATA 25,30,125, 23,181,125
109 DATA 25,30,125, 23,181,125
110 DATA 21,31,250
111 DATA 21,31,125, 23,181,125
112 DATA 25,30,125, 23,181,125
113 DATA 21,31,250, 21,31,250
114 DATA 21,31,125, 18,209,125
115 DATA 16,195,125,15,210,125
116 DATA 16,195,125,15,210,125
117 DATA 14,24,250
118 DATA 14,24,125, 15,210,125
119 DATA 16,195,125,18,209,125
120 DATA 21,31,250, 21,31,250
121 DATA 21,31,125, 18,209,125
122 DATA 16,195,125,15,210,125
123 DATA 16,195,125,15,210,125
124 DATA 14,24,250
125 DATA 14,24,125, 15,210,125
126 DATA 16,195,125,15,210,125
127 DATA 14,24,250, 14,24,250
128 DATA -1,-1,-1
```

Most a következőképpen oldottuk meg a fejezet elején kitűzött feladatokat:

1. A megszólaló hang frekvenciáját nem a P paraméterrel, hanem az általa meghatározott F felső és A alsó byte beolvasásával adjuk meg. (Ezeket az értékeket szintén az 1. táblázat alapján határozhatjuk meg.) Így valamivel több számot kell a DATA sorokba írnunk. Megéri, mert a program futása gyorsabb lesz, hiszen az előbb a P paramétert zenélés közben kellett F-re és A-ra szétvágni.

2a. A kitartási idő kezelése hasonlóképpen történik, mint az előző programban.

2b. A hang elengedésére szánt idő lényegében nulla. Ez érthető, hiszen a kitartási szintet, és az elengedési időt meghatározó szám (S, ill. R) nulla.

Egy újabb különbség: az előbb FOR ciklussal olvastuk be az adatokat, most pedig addig olvasunk, míg a beolvasott szám negatív nem lesz. Így a 128-as sornak az a szerepe, hogy jelzi a beolvasás végét.

Azok kedvéért, akik a sok DATA sortól már most megijednek, ismét megjegyzem, hogy begépelés előtt érdemes alaposan végignézni a programot, hátha találunk többé-kevésbé azonos sorokat. (Ez a következő példákra fokozottan érvényes.)

3. példa

A program a régi Illés együttes egy kevéssé ismert számát szólaltatja meg. Címe: "Üzenet Eddynek", s azon kevés Illés számok egyike, melyekben a fiúk nem énekelnek, a főszerep a gitáré és a szaxofoné.

A program elkészítése azzal kezdődött, hogy gitáron eljátszottam majd lekottáztam a számot. Ezt követte a hangfrekvenciáknak megfelelő paraméterek meghatározása és beírásuk a DATA sorokba, valamint a program inicializáló és vezérlő részének megírása. Az utolsó művelet a kitartási időt meghatározó paraméterek változtatása volt egészen addig, míg a program mostani formája ki nem alakult.

(Ha az Olvasónak megvan az eredeti szám, próbálja ki a program zenéjét és a hangfelvételt egyszerre elindítani. Ehhez érdemes két új sorral bővíteni programunkat:

```
33 PRINT "OK"
34 GET X$ : IF X$ = "" THEN 34
```

Így ugyanis az OK felirat megjelenése után egy billentyű lenyomásával tetszőleges pillanatban elindíthatjuk a zenét.)

```
1 REM ILLES:UZENET EDDYNEK /CSIKOS ZS./
2 PRINT"♥"
3 POKE53280,0:POKE53281,0
6 DIM A(12,2),B(38,2),C(18,2),D(35,2),E(27,2)
8 S=54272:TI$="000000"
9 FORI=0TO24:POKES+I,0:NEXT
10 PRINT"◣ ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●"
11 PRINT"■ ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●"
12 PRINT"■    ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●"
13 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
14 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
15 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
16 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●●●"
17 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●●●"
18 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●■●●"
19 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
20 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
21 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
22 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
23 PRINT"■    ●■●●●●■■●●●■●●●■●●●■●●●■●●●●■●●●●■●●●●●●■●"
24 PRINT"■●●◣■●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●◣■●●●"
26 FORI=0TO12:FORJ=0TO2:READA(I,J):NEXTJ:NEXTI
27 FORI=0TO38:FORJ=0TO2:READB(I,J):NEXTJ:NEXTI
28 FORI=0TO18:FORJ=0TO2:READC(I,J):NEXTJ:NEXTI
29 FORI=0TO35:FORJ=0TO2:READD(I,J):NEXTJ:NEXTI
30 FORI=0TO27:FORJ=0TO2:READE(I,J):NEXTJ:NEXTI
```

```
31 POKES+24,5
32 POKES+6,249:POKES+13,247
35 T=TI
37 REM **************
41 FORI=0TO12
42 POKES+1,A(I,0):POKES,A(I,1):POKES+4,33
43 T=T+A(I,2)
44 IFT>TI THEN 44
45 POKES+4,32
46 NEXT
47 REM **************
50 FORJ=1TO2
51 FORI=0TO38
52 POKES+1,B(I,0):POKES,B(I,1):POKES+4,33
53 T=T+B(I,2)
54 IFT>TI THEN 54
55 POKES+4,32
56 NEXTI,J
57 REM **************
60 FORJ=1TO2
61 FORI=0TO18
62 POKES+1,C(I,0):POKES,C(I,1):POKES+4,33
63 T=T+C(I,2)
64 IFT>TI THEN 64
65 POKES+4,32
66 NEXTI,J
67 REM **************
71 FORI=0TO35
72 POKES+1,D(I,0):POKES,D(I,1):POKES+4,33
73 T=T+D(I,2)
74 IFT>TI THEN 74
75 POKES+4,32
76 NEXTI
77 REM **************
81 FORI=3TO38
82 POKES+1,B(I,0):POKES,B(I,1):POKES+4,33
83 T=T+B(I,2)
84 IFT>TI THEN 84
85 POKES+4,32
86 NEXTI
87 IFV=1THEN 110
88 REM **************
91 FORI=0TO2
92 POKES+1,B(I,0):POKES,B(I,1):POKES+4,33
93 T=T+B(I,2)
94 IFT>TI THEN 94
95 POKES+4,32
96 NEXTI
97 REM **************
100 FORJ=1TO2
101 FORI=0TO27
102 IFE(I,0)THENPOKES+1,E(I,0):POKES,E(I,1):POKES+4,33
103 T=T+E(I,2)
104 IFT>TI THEN 104
105 POKES+4,32
106 NEXTI,J
107 V=1:GOTO60
108 REM **************
```

```
110 FORJ=1TO3
111 FORI=19TO27:K=I+12
112 POKES+1,E(I,0):POKES,E(I,1):POKES+4,33
113 POKES+8,B(K,0):POKES+7,B(K,1):POKES+11,33
114 T=T+E(I,2)
115 IFT>TI THEN 115
116 POKES+4,32:POKES+11,32
117 NEXTI,J
300 DATA 6,232,20 ,5,123,120
301 DATA 5,123,10, 6,39,10, 6,232,10, 7,81,130
302 DATA 6,133,5 , 6,232,20 ,5,123,120
303 DATA 5,123,10, 6,39,10, 6,232,10, 7,81,130
304 REM ****
305 DATA 7,81,10, 6,232,10, 6,39,10
306 DATA 5,123,20, 5,123,20, 6,232,20, 8,55,20
307 DATA 10,247,15, 10,247,10, 9,196,20, 9,56,10, 8,55,20
308 DATA 7,81,20, 7,81,20, 5,123,20, 6,39,20, 6,133,15, 6,133,10
309 DATA 7,81,20, 7,81,10, 6,133,10, 6,39,10
310 DATA 5,123,20, 5,123,20, 6,232,20, 8,55,20
311 DATA 10,247,20, 10,247,10, 9,196,20, 9,56,10, 8,55,20
312 DATA 9,196,20, 9,196,20, 8,180,20, 8,180,20
313 DATA 8,55,10, 7,81,10, 8,55,10, 7,81,20
323 REM ****
324 DATA 0,0,5
325 DATA 6,133,20, 6,133,10, 7,81,10, 8,55,10, 7,81,10, 6,133,20
326 DATA 7,81,20, 7,81,10, 7,81,20, 6,133,10, 7,81,20
327 DATA 8,180,60, 0,0,20
328 DATA 6,133,20, 7,81,10, 6,133,20, 7,81,10, 6,133,15
339 REM ***
340 DATA 7,81,20, 7,81,10, 8,55,10, 9,56,10, 8,55,10, 7,81,20
341 DATA 8,55,20, 8,55,10, 8,55,20, 7,81,10,8,55,20
342 DATA 9,196,40, 9,196,20
343 DATA 8,55,10, 8,180,10, 8,180,10, 8,180,20, 8,180,10, 8,180,20
344 DATA 8,55,20, 8,55,10, 8,55,20, 8,55,10, 8,55,20
345 DATA 8,180,20, 8,180,10, 8,180,20, 8,55,10, 8,180,10, 8,55,10
346 DATA 7,81,40, 0,0,40, 6,133,20, 6,39,20, 5,123,20, 0,0,5
350 REM *********
351 DATA 0,0,20
352 DATA 32,219,50, 31,3,10, 32,219,10, 36,225,20, 32,219,10, 0,0,40
353 DATA 36,227,50, 39,18,10, 36,225,10, 32,219,10, 36,225,20, 0,0,50
354 DATA 32,219,50, 31,3,10, 32,219,10, 36,225,20, 32,219,10, 0,0,40
355 DATA 26,20,20, 26,20,20, 34,207,20, 34,207,20
356 DATA 32,219,10, 29,69,10, 32,219,10, 29,69,20, 0,0,30
```

A felhasznált jelek:

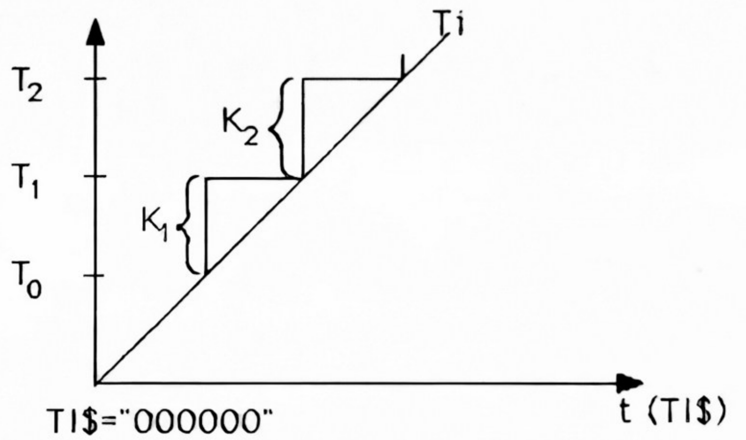
 = SHIFT + CLR/HOME  = CTRL + 4

 = CTRL + 3  = CTRL + 7  = CTRL + 8

A 2. sor törli a képernyőt, a 3. sor a keret- és háttérszínt állítja be feketére. A 10-24 sorok hatására a képernyőn nagy ILLÉS felirat jelenik meg. E program két fő elemben különbözik az előbbiektől. Az egyik, hogy most nem a dallam lejátszása közben olvassuk be a hangokat meghatározó adatokat, hanem a program elején, és <u>tömbökben</u> tároljuk azokat. Háromdimenziós tömböket használunk: az első két dimenzió a frekvenciát meghatározó byte-ok értékeit tartalmazza (ez volt az előbb F és A), míg a harmadik a kitartási időt határozza meg (ez volt az előbb K). A tömbök

dimenzionálása a 6. sorban történik meg, és ezeket a 26-30. sorokban töltjük fel adatokkal. A tömbök használatának hátránya, hogy a programot futtatva nem szólal meg azonnal a hang, várakoznunk kell egy keveset, hiszen a tömbök feltöltése időt igényel. Hallatlan előnye viszont a módszernek, hogy a dallamrészek ismétléseinek programozását jelentősen megkönnyíti. Az előző példákban a dallam elejére egyszerűen visszaugorhattunk (RESTORE utasítás), a részletek ismétlését azonban csak bizonyos DATA-sorok többszöri beírásával, vagy a DATA-pointer megfelelő átállításával tudtuk elérni. Az utóbbi igen nehézkes, a DATA sorok ismétlése pedig fölöslegesen szaporítja, s néha áttekinthetetlenné teszi az adatokat. Hiba elkövetése után is többször kell ugyanazt kijavítani.

Programunk másik újdonsága, hogy a kitartási időt nem üres ciklussal, hanem a számítógép belső órájával, a TI és a TI$ segítségével programozzuk. Ennek lényege a következő: a program elején (8.sor) nullára állítjuk a TI$-t, hogy a gépnek minél kisebb számokkal kelljen dolgoznia. Ekkor TI értéke is nullára változik. A dallam lejátszása előtt - és lehetőleg a tömbök feltöltése után - egy változóhoz (T) hozzárendeljük a TI pillanatnyi értékét. A kitartási időt meghatározó paramétert hozzáadjuk ehhez a változóhoz, és a megszólaló hangot addig tartjuk a kitartási szinten, míg az időt mutató TI el nem éri ezt az értéket (35.,43.,44. sor, stb.).



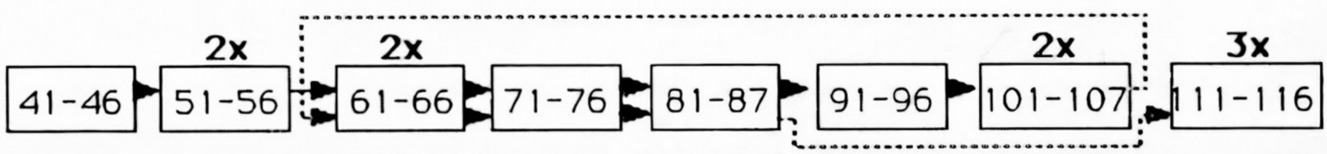
33. ábra

ahol:

T_i - a T változó értéke az i módosítás után

K_i - i megszólaló hang kitartását meghatározó paraméter — jelen esetben a tömbök harmadik dimenziója.

A dallamban történő ugrások vázlata:
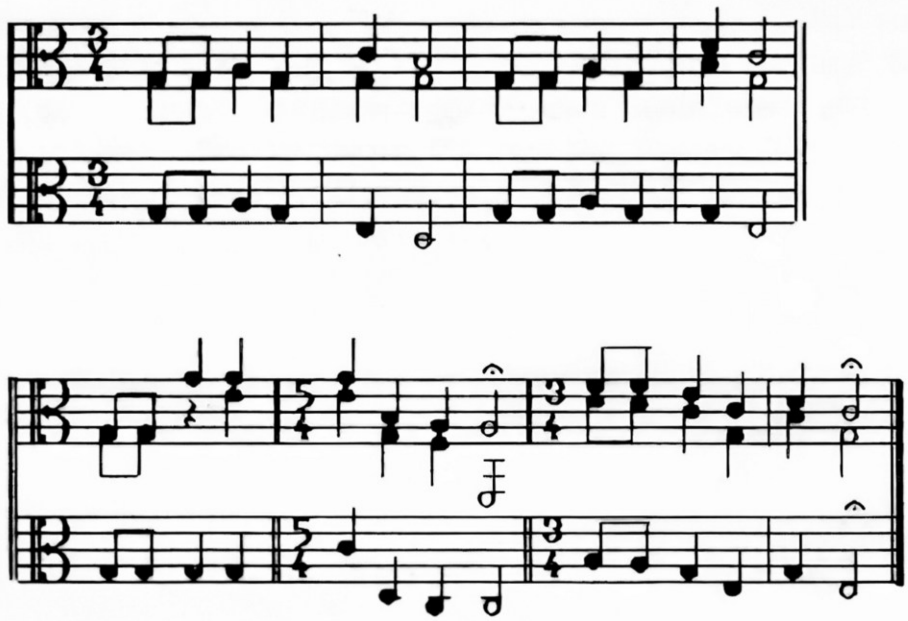


Az ábrából jól kitűnik, hogy tényleg érdemes volt tömböket használnunk. A program gépelését megkönnyíti, hogy a 37-108-as sorokig a REM ****-okkal kezdődő egységek között minimális a különbség.

A programfutás befejező szakaszában a két "hangszer" egyszerre szólal meg (110-117).

A következő példánkban már mind a három hanggenerátort használni fogjuk.

4. *példa*

Valószínűleg az olvasók többsége énekelte már a "Happy Birthday" c. angol születésnapi dalocskát. A dalt Kabdebó Sándor írta át zongorára, s kérésemre ebből az átiratból választott ki három szólamot. Programunk lényegében a 2. példában szereplő programhoz hasonlóan működik, három szólamban. (A 15-ös sorban szereplő ♥ = SHIFT+CLR, és Q = CRSR ∇) Gépeléskor ismét érdemes az azonos DATA-sorokra figyelni!



```
10 REM HAPPY BIRTHDAY TO YOU
11 REM /KABDEBO SANDOR & CSIKOS ZS/
15 PRINT"♥QQQQQQQQ"TAB(9);"HAPPY BIRTHDAY TO YOU!"
20 S=54272
25 FORI=0TO24:POKES+I,0:NEXT
30 POKES+24,15
35 POKES+5,9:POKES+12,9:POKES+19,9
40 POKES+3,4:POKES+10,7
45 READF1,A1
50 READF2,A2
55 READF3,A3
60 READK
65 IFF1<0 THEN RESTORE:V=V+1:GOTO45
70 IF V=2THEN END
75 POKES+1,F1:POKES+8,F2:POKES+15,F3
80 POKES,A1:POKES+7,A2:POKES+14,A3
85 POKES+4,65:POKES+11,65:POKES+18,33
90 FORI=1TOK:NEXT
95 POKES+4,32:POKES+11,32:POKES+18,32
99 GOTO45
300 DATA 12,143,12,143,12,143,150
301 DATA 12,143,12,143,12,143,150
302 DATA 14,24,14,24,14,24,400
303 DATA 12,143,12,143,12,143,400
304 DATA 16,195,12,143,10,143,400
305 DATA 15,210,12,143,9,104,900
```

```
310 DATA 12,143,12,143,12,143,150
311 DATA 12,143,12,143,12,143,150
312 DATA 14,24,14,24,14,24,400
313 DATA 12,143,12,143,12,143,400
314 DATA 18,209,15,210,12,143,400
315 DATA 16,195,12,143,10,143,900
320 DATA 12,143,12,143,12,143,150
321 DATA 12,143,12,143,12,143,150
322 DATA 25,30,0,0,12,143,400
323 DATA 25,30,21,31,12,143,400
324 DATA 25,30,21,31,16,195,400
325 DATA 15,210,12,143,9,104,400
326 DATA 14,24,11,48,8,97,400
327 DATA 14,24,5,152,8,97,1000
330 DATA 22,96,18,209,14,24,150
331 DATA 22,96,18,209,14,24,150
332 DATA 21,31,16,195,12,143,400
333 DATA 16,195,12,143,10,143,400
334 DATA 18,209,15,210,12,143,400
335 DATA 16,195,12,143,10,143,1000
340 DATA-1,0,0,0,0,0,0
```

5. *példa*

A türelmetlenebb olvasóknak már bizonyára elegük van a rengeteg DATA sorból. Arra kérem őket, hogy ne hagyják félbe emiatt a könyv olvasását! Gondolkodjunk el inkább azon, hogy hogyan adhatnánk meg egyszerűbben egy-egy dallam hangjait.

Mi lenne, ha egy kétdimenziós segédtömböt (HB) feltöltenénk a skálahangok frekvenciáját meghatározó felső, ill. alsó byte-okkal? Így a megszólaltatni kívánt hangok magasságát nem a frekvenciákhoz tartozó paraméterekkel kell majd megadnunk, hanem egymáshoz viszonyított helyzetükkel. Ha ezt megtesszük, akkor pl. a C-4, F-4, D-4, D#-4 hangmagasságok megadásához elegendő lesz a (0,5,2,3) számokat beírni a DATA sorokba, az eddigi (4455,5947,5001,5298) vagy (17,103;23,59;19,137;20,178) helyett.

A különbség valóban jól érzékelhető. Sőt az is nyilvánvaló, hogy ennél rövidebben már nem lehet hangmagasságot megadni.

A következő kérdés is az, hogy tudunk-e még valamit egyszerűsíteni - és a válasz természetesen most is igen. Az Illés-programnál láthattuk, hogy a zene megszólaltatását biztosító programrészek között milyen csekély a különbség. Hogyan lehetne itt munkát megtakarítani? A megoldás: egy szubrutint hozunk létre:

```
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1)
30 FORI=1TO95
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
900 END
1000 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006 IFKTHENPOKES+4,H1+1
1008 IFLTHENPOKES+11,H2+1
1010 IFMTHENPOKES+18,H3+1
1012 FORT=1TON:NEXT
1014 IFKTHENPOKES+4,H1
1016 IFLTHENPOKES+11,H2
1018 IFMTHENPOKES+18,H3
1020 RETURN
```

Figyeljük meg, hogy a HB tömb feltöltésekor nem használunk kerekítést. Ennek az a célja, hogy ne nőjön tovább a program elindítása és a zenélés megkezdése közötti idő. A kerekítés elhgyása miatt a 262.8372 egy kicsit magasabb a valódi értéknél, így a HB tömb értékei és az igazi értékek között minimális a különbség. A HB (1,1) és HB (1,0) értékek C-0-hoz tartoznak.

A szubrutinban a következő paraméterek szerepelnek:
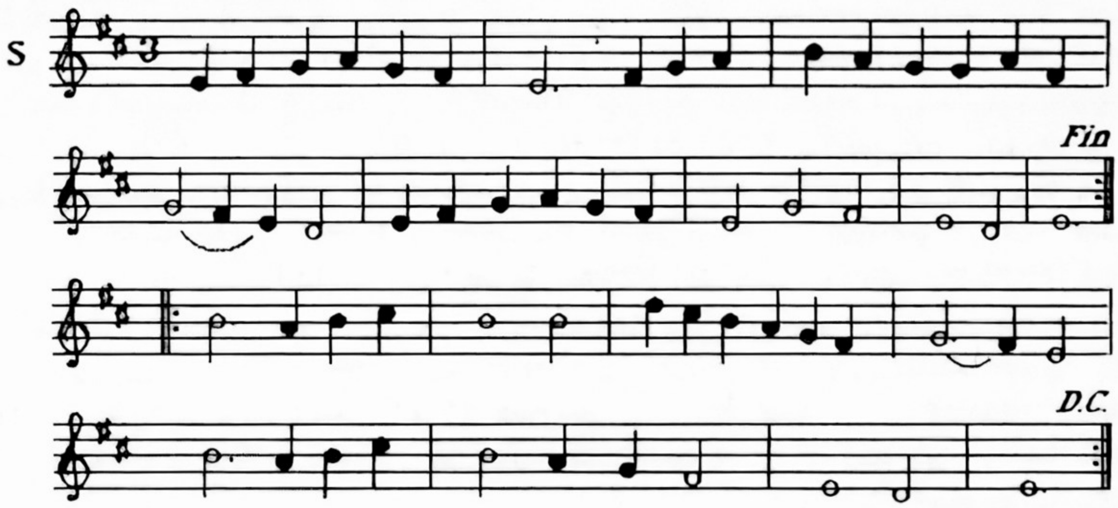
K:	az első hanggenerátoron megszólaló hang HB szerinti sorszáma (pl. E-3 esetén 41)
L:	a második hanggenerátoron megszólaló hang sorszáma
M:	a harmadik hanggenerátoron megszólaló hang sorszáma
N:	a kitartási idő hosszát meghatározó üres ciklus végértéke
H1,H2,H3:	az egyes hangokhoz tartozó hullámforma.
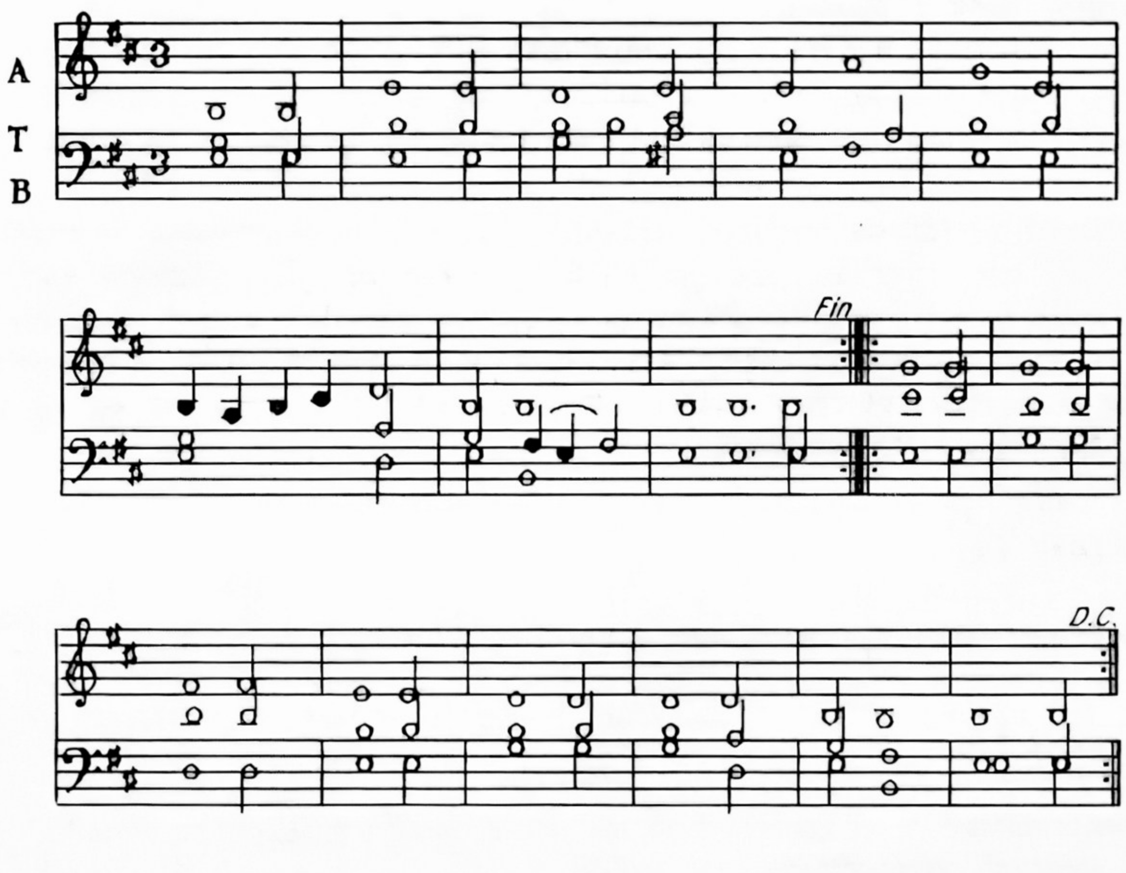
Az IF (ARITMETIKAI KIFEJEZÉS) THEN... utasítás esetén a THEN ág akor hajtódik végre, ha a kifejezés értéke nem nulla. Így például, ha K értéke 0, akkor az 1000-es utasításról a program azonnal továbbugrik az 1002-esre, s így megszabadulunk a felesleges kattogásoktól.

Érdemes ezt a programot jelenlegi formájában, önállóan is kimenteni, mert a további példák közül több is erre fog épülni. Nem kell elkeseredni, ha az eddig elmondottakból nem minden érthető, a példák majd mindenre magyarázatot adnak.

6. példa

A program egy reneszánsz dallamot szólaltat meg. A dal címe Tourdion, szerzője ismeretlen. 1530-ban publikálta Pierre Attaingnant. A dallal egészen véletlenül ismerkedtem meg. Egy lemez-leértékelés alkalmával megvettem a Hortus Musicus egyik nagylemezét, melyen ez a mű is szerepel hangszeres feldolgozásban. Később egy kórusfesztiválon hallottam ismét, ahol egy pármai énekkar adta elő. A pármaiak nagy sikert arattak, remélem a dal a C-64 előadásában is tetszeni fog hallgatóinak.





Mivel a négy szólam megszólaltatását BASIC-ben igen nehéz lenne programozni, ezért a dalból az alt szólamát elhagyjuk.

Nézzük meg, hogy hogyan tudjuk meghatározni pl.a szoprán hangjait! Két kereszt az előjegyzés, tehát D a dó, s így az első hang (e) ré-nek tekinthető. A kottában szereplő legmélyebb hang a d , ezért ezt választjuk 0-nak, és a többi hang sorszámát ehhez fogjuk viszonyítani. Így a kezdőhang sorszáma 2 lesz, és az első ütem kódolása: (r,m,f,s,f,m) --> (2,4,5,7,5,4)

A kottában szereplő legkisebb hangjegyérték a negyed, ezt tekinthetjük egységnek. Ha negyednél hosszabb hanggal találkozunk, akkor el kell döntenünk, hogy hogyan adjuk meg. (Pl. A második ütem első hangja 3/4, vagyis háromszor annyi ideig kell kitartanunk, mint az előbbieket.) Két dolgot tehetünk: vagy háromszor adjuk meg a hangot (2,2,2) és az ADSR értékét úgy választjuk meg, hogy a három megszólaltatás közötti különbség elmosódjon, vagy pedig egyszer adjuk csak meg, és utána két negyedig K értékét nullára állítjuk. Ha az S és az R értékét is megfelelően állítjuk be, akkor így a hang fokozatosan halkulva három negyedérték időtartamáig szól. E példában az utóbbi módszert választottam, mert így könnyebb különbséget tenni pl. egy fél-, és két ugyanolyan magasságú negyedhang között.

Az adatok beolvasása után tömbjeinket már a valódi sorszámokkal kell feltölteni, nem pedig a relatívakkal. A szoprán esetében például – ahol a d^1-et (D-4) választottuk nullának –, a beolvasott értékhez 51-et hozzá kell adnunk. Amennyiben a beolvasott szám negatív (-1), a tömb megfelelő eleme nulla marad; ezt az 55-57. sorok biztosítják. Hasonlóan történik a tenor és a basszus adatainak megadása és tömbökké alakításuk is.

Most már csak a szubrutin hívását kell megnéznünk. Tekintsük pl. a 95-110 sorokat! A 95. sorban a három hullámformát állítjuk be. A 100-110 sorok utasításai az éppen aktuális szoprán, tenor és basszus hang sorszámát helyezik el a K, L, illetve M változókba. A programot kipróbálás után futtassuk le különböző N értékek mellett is!

A kottától eltértünk néhány tekintetben: A mű eredetileg három szakaszra tagolódik: az első rész kétszer hallható, a második rész szintén kétszer, végül megismétlődik az első rész. Programunk az utóbbi ismétlést elhagyja, az első két részt azonban négyszer játssza el, mégpedig más és más hangszíneken, mintegy különböző hangszerereket imitálva. Egy másik eltérés a kottától, hogy az egyik hangot fél hanggal mélyebbnek választottuk.

M Feladat 13.

A gyakorlás kedvéért, keressük meg melyik ez a hang!

M Feladat 14.
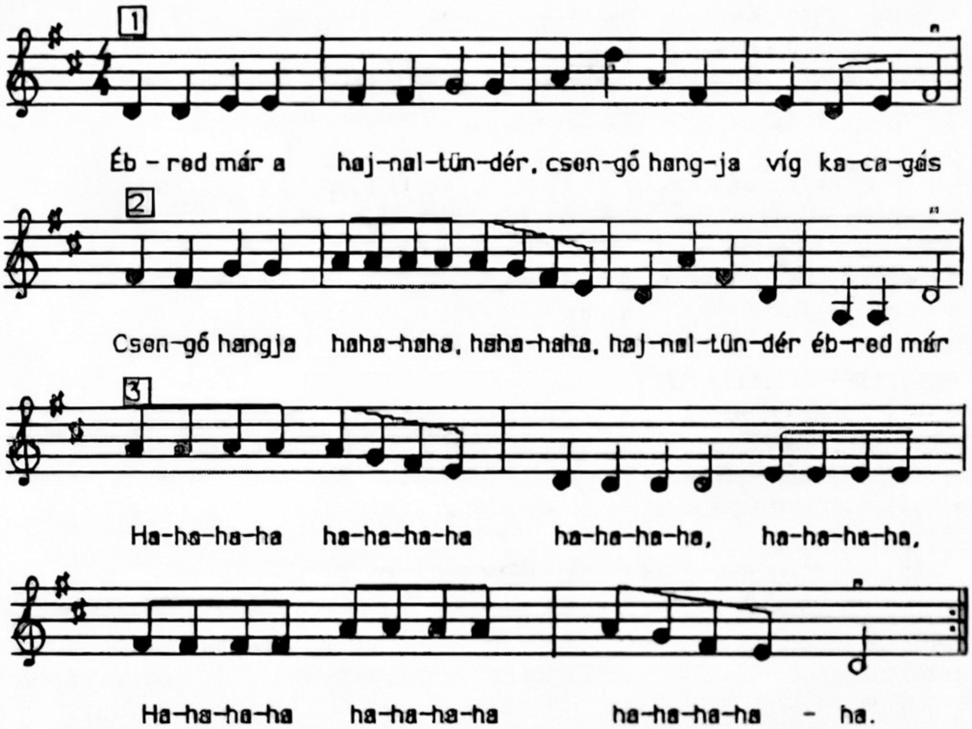
Hogyan lehetne a program indítása, és az első hang megszólalása közti időt a program minimális átalakításával lecsökkenteni?

```
1 REM TOURDION
2 REM SZAMITOGEPRE ATIRTA:CSIKOS ZS.
3 PRINT"{CLR}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}"TAB(16)"TOURDION"
4 PRINT"{DOWN}"TAB(9)"ISMERETLEN SZERZO MUVE"
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1)
30 FORI=1TO95
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
50 DIM S(192),T(192),B(195),S1(192)
54 REM ****
55 FORI=1TO192:READX:IFX<0THEN 57
56 S(I)=X+51:S1(I)=S(I)+12
57 NEXT
65 FORI=1TO192:READX:IFX<0THEN 67
66 T(I)=X+41
67 NEXT
75 FORI=1TO192:READX:IFX<0THEN 77
76 B(I)=X+36
77 NEXT
78 REM ****
90 POKES+24,15
91 POKES+6,250:POKES+13,172:POKES+20,172
92 POKES+10,1:POKES+17,7
94 REM ****
95 H1=32:H2=32:H3=64
100 FORI=1TO192
105 K=S(I):L=T(I):M=B(I):N=100:GOSUB1000
110 NEXTI
111 REM ***
115 H1=16:H2=16:H3=16
120 FORI=1TO192
125 K=S(I):L=T(I):M=B(I):N=100:GOSUB1000
130 NEXTI
131 REM ***
135 H1=16:H2=32:H3=64
140 FORI=1TO192
145 K=S1(I):L=T(I):M=B(I):N=100:GOSUB1000
150 NEXTI
151 REM ***
155 H1=32:H2=32:H3=64
160 FORI=1TO192
165 K=S(I):L=T(I):M=B(I):N=100:GOSUB1000
170 NEXTI
200 REM SZOPRAN
201 DATA 2,4,5,7,5,4, 2,-1,-1,4,5,7
202 DATA 9,7,5,5,7,4, 5,-1,4,2,0,-1
203 DATA 2,4,5,7,5,4, 2,-1,5,-1,4,-1
204 DATA 2,-1,-1,-1,0,-1, 2,-1,-1,-1,-1,-1
205 REM **
206 DATA 2,4,5,7,5,4, 2,-1,-1,4,5,7
207 DATA 9,7,5,5,7,4, 5,-1,4,2,0,-1
208 DATA 2,4,5,7,5,4, 2,-1,5,-1,4,-1
209 DATA 2,-1,-1,-1,0,-1, 2,-1,-1,-1,-1,-1
210 REM **
211 DATA 9,-1,-1,7,9,11, 9,-1,-1,-1,9,-1
```

```
212 DATA 12,11,9,7,5,4, 5,-1,-1,4,2,-1
213 DATA 9,-1,-1,7,9,10, 9,-1,7,5,4,-1
214 DATA 2,-1,-1,-1,0,-1, 2,-1,-1,-1,-1,-1
215 REM **
216 DATA 9,-1,-1,7,9,11, 9,-1,-1,-1,9,-1
217 DATA 12,11,9,7,5,4, 5,-1,-1,4,2,-1
218 DATA 9,-1,-1,7,9,10, 9,-1,7,5,4,-1
219 DATA 2,-1,-1,-1,0,-1, 2,-1,-1,-1,-1,-1
240 REM TENOR
241 DATA 3,-1,-1,-1,0,-1, 7,-1,-1,-1,7,-1
242 DATA 7,-1,-1,-1,8,-1, 7,-1,-1,-1,5,-1
243 DATA 7,-1,-1,-1,7,-1, 3,-1,-1,-1,5,-1
244 DATA 3,-1,2,0,2,-1, 0,-1,-1,-1,0,-1
245 REM **
246 DATA 3,-1,-1,-1,0,-1, 7,-1,-1,-1,7,-1
247 DATA 7,-1,-1,-1,8,-1, 7,-1,-1,-1,5,-1
248 DATA 7,-1,-1,-1,7,-1, 3,-1,-1,-1,5,-1
249 DATA 3,-1,2,0,2,-1, 0,-1,-1,-1,0,-1
250 REM **
251 DATA 12,-1,-1,-1,12,-1, 10,-1,-1,-1,10,-1
252 DATA 10,-1,-1,-1,10,-1, 7,-1,-1,-1,7,-1
253 DATA 7,-1,-1,-1,7,-1, 7,-1,-1,-1,5,-1
254 DATA 3,-1,2,-1,-1,-1, 0,-1,-1,-1,0,-1
255 REM **
256 DATA 12,-1,-1,-1,12,-1, 10,-1,-1,-1,10,-1
257 DATA 10,-1,-1,-1,10,-1, 7,-1,-1,-1,7,-1
258 DATA 7,-1,-1,-1,7,-1, 7,-1,-1,-1,5,-1
259 DATA 3,-1,2,-1,-1,-1, 0,-1,-1,-1,0,-1
260 REM BASSZUS
261 DATA 5,-1,-1,-1,5,-1, 5,-1,-1,-1,5,-1
262 DATA 8,-1,12,-1,10,-1, 5,-1,7,-1,-1,-1
263 DATA 5,-1,-1,-1,5,-1, 5,-1,-1,-1,3,-1
264 DATA 5,-1,0,-1,-1,-1, 5,-1,-1,-1,-1,-1
265 REM **
266 DATA 5,-1,-1,-1,5,-1, 5,-1,-1,-1,5,-1
267 DATA 8,-1,12,-1,10,-1, 5,-1,7,-1,-1,-1
268 DATA 5,-1,-1,-1,5,-1, 5,-1,-1,-1,3,-1
269 DATA 5,-1,0,-1,-1,-1, 5,-1,-1,-1,-1,-1
270 REM **
271 DATA 5,-1,-1,-1,5,-1, 8,-1,-1,-1,8,-1
272 DATA 3,-1,-1,-1,3,-1, 5,-1,-1,-1,5,-1
273 DATA 8,-1,-1,-1,8,-1, 8,-1,-1,-1,3,-1
274 DATA 5,-1,0,-1,-1,-1, 5,-1,-1,-1,5,-1
275 REM **
276 DATA 5,-1,-1,-1,5,-1, 8,-1,-1,-1,8,-1
277 DATA 3,-1,-1,-1,3,-1, 5,-1,-1,-1,5,-1
278 DATA 8,-1,-1,-1,8,-1, 8,-1,-1,-1,3,-1
279 DATA 5,-1,0,-1,-1,-1, 5,-1,-1,-1,5,-1
900 END
1000 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006 IFKTHENPOKES+4,H1+1
1008 IFLTHENPOKES+11,H2+1
1010 IFMTHENPOKES+18,H3+1
1012 FORT=1TON:NEXT
1014 IFKTHENPOKES+4,H1
1016 IFLTHENPOKES+11,H2
1018 IFMTHENPOKES+18,H3
1020 RETURN
```

7. példa

Ez a program is az 5. példában megismert segédtömbre és szubrutinra épül. Most azonban nem töltjük fel a teljes HB tömböt, hanem csak azon elemei kapnak értéket, amelyekre ténylegesen szükségünk lesz a program futásakor (30-as sor). A megszólaló dal egy népszerű háromszólamú angol kánon, melynek címe "Ébred a hajnal". (A kotta Kerényi György magyar szövegét is tartalmazza.)



A kottában a bekeretezett számok az egyes szólamok belépését mutatják, a ⊓ jelek pedig azokat a hangokat jelzik, ahol a szólamok befejezik az éneklést. Mivel a kánon három szólamában ugyanaz a dallam hallható, a hangokat meghatározó adatokat elegendő egyszer beírni a DATA sorokba. A program egyik érdekessége a szólamok hangjait tartalmazó tömbök (A,B,C) feltöltése (50-63).

A megszólaló hangok hangszínét úgy kellett megválasztani, hogy jól érezhető legyen a szólamok fáziseltolódása. Arra is figyelni kellett, hogy N értékét kezdetben, - amikor csak az első, ill. csak az első két szólam szól, - nagyobbra állítsuk be, mint amikor már mind a hármat egyszerre halljuk. (N = 130, N = 120, ill. N = 100)

```
1 REM EBRED MAR A ... /3 SZOLAMU KANON/
2 REM     /CSIKOS ZS./
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1),A(96),B(96),C(96)
30 FORI=46TO63
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
42 POKES+24,15
44 POKES+6,251:POKES+13,250:POKES+19,11
45 POKES+2,150:POKES+17,15
50 FORI=1TO32:READA(I):IFA(I)<0THENA(I)=0:GOTO52
51 A(I)=A(I)+46:B(I+32)=A(I):C(I+64)=A(I)
52 NEXT
55 FORI=33TO64:READA(I):IFA(I)<0THENA(I)=0:GOTO57
56 A(I)=A(I)+46:B(I+32)=A(I):C(I-32)=A(I)
57 NEXT
60 FORI=65TO96:READA(I):IFA(I)<0THENA(I)=0:GOTO63
61 A(I)=A(I)+46:B(I-64)=A(I):C(I-32)=A(I)
63 NEXT
70 H1=64:H2=16:H3=32
72 FORI=1TO32:K=A(I):N=130:GOSUB1000:NEXT
74 FORI=33TO64:K=A(I):L=B(I):N=120:GOSUB1000:NEXT
76 FORI=65TO96:K=A(I):L=B(I):M=C(I):N=100:GOSUB1000:NEXT
78 POKES+3,7:POKES+19,7:POKES+6,171
80 FORI=1TO96:K=A(I):L=B(I):M=C(I):N=100:GOSUB1000:NEXT
82 H2=16:H3=64:POKES+19,0:POKES+20,139
84 FORI=1TO95:K=A(I):L=B(I):M=C(I):N=100:GOSUB1000:NEXT
86 POKES+6,252
88 K=A(96):L=B(96):M=C(96):N=100:GOSUB1000
200 DATA 5,-1,5,-1,7,-1,7,-1
202 DATA 9,-1,9,-1,10,-1,10,-1
204 DATA 12,-1,17,-1,12,-1,9,-1
206 DATA 7,-1, 5, 7,9,-1,-1,-1
208 DATA 9,-1, 9,-1,10,-1,10,-1
210 DATA 12,12,12,12,12,10,9,7
212 DATA 5,-1,12,-1,9,-1,5,-1
214 DATA 0,-1, 0,-1,5,-1,-1,-1
216 DATA 12,12,12,12,12,10,9,7
218 DATA 5,5,5,5,7,7,7,7
220 DATA 9,9,9,9,12,12,12,12
222 DATA 12,10,9,7,5,-1,-1,-1
900 END
1000 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006 IFKTHENPOKES+4,H1+1
1008 IFLTHENPOKES+11,H2+1
1010 IFMTHENPOKES+18,H3+1
1012 FORT=1TON:NEXT
1014 IFKTHENPOKES+4,H1
1016 IFLTHENPOKES+11,H2
1018 IFMTHENPOKES+18,H3
1020 RETURN
```

8. példa

A három hanggenerátor nemcsak akkor van hasznunkra, ha többszólamú dalt akarunk számítógéppel megszólaltatni. Ha például gitáron játszunk, akkor természetes, hogy egy-egy húr nem fog azért elhallgatni, mert egy újabbat is megpendítünk. A gitárnak ugyan hat húrja van, de a három hanggenerátorral is elég jól lehet utánozni. Ezt mutatja be a következő program, mely Al Di Meola: "Mediterranean Sundance" c. számának kezdetét játssza el. Az egyes húrok megszólalásának sorrendjénél azt vettem figyelembe, ahogy én szoktam játszani ezt a számot, - de ezen bárki tetszése szerint változtathat.

Tekintsük át a programot! Az elején ugyanúgy töltjük fel a HB tömböt, mint a korábbiakban. Ezen kívül négy másik tömböt is használunk, érdemes ezeket is megvizsgálnunk. Az A és a D tömb az egymás után megszólaló hangok adatait tartalmazzák. A B tömb elemei azt határozzák meg, hogy melyik húr szól, vagyis, hogy az S., S+7., illetve az S+14. byte-ba kell-e beírnunk a megfelelő frekvenciaértéket. Végül az F tömb egy akkord három hangját adja meg. (A programban szereplő A,B,C változókat talán szerencsésebb lett volna más betűkkel jelölni, de remélem, így sem zavaró a használatuk.)

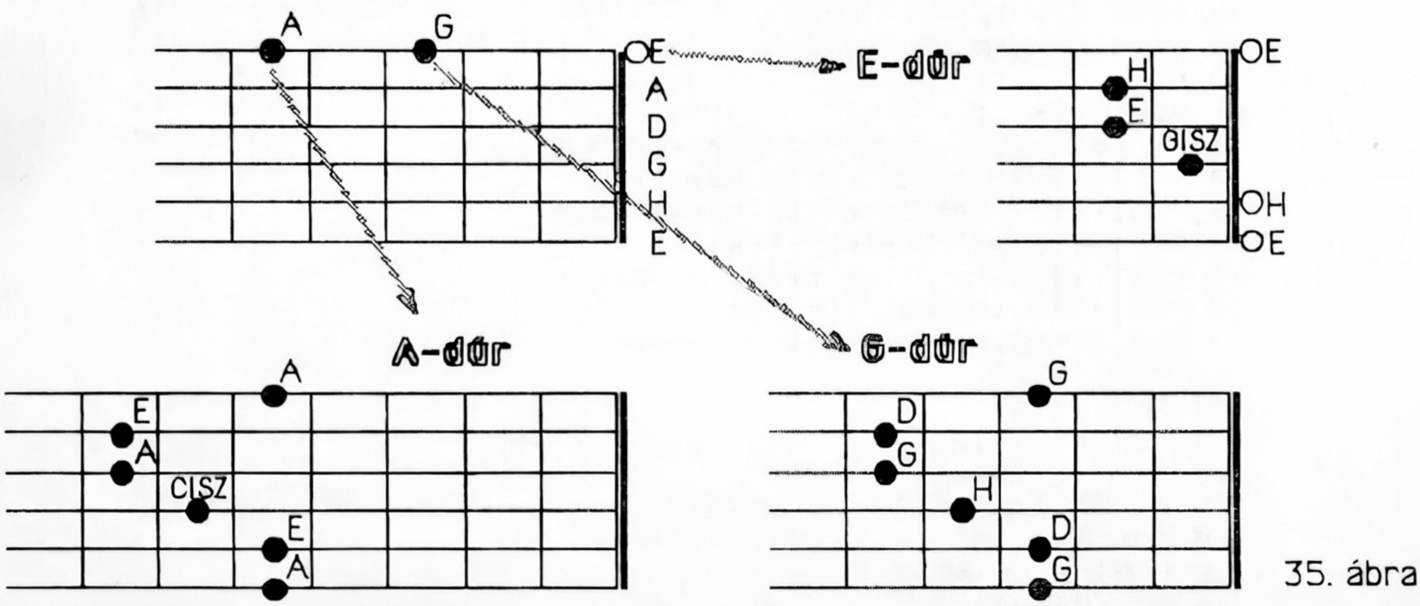
A zenélő rész három kisebb egységből tevődik össze. Az első (102-118) és a második (120-132) formailag közel azonos, míg a harmadik (139-148) - mely az akkordokat szólaltatja meg -, különbözik ezektől. Figyeljük meg az ADSR értékek alakulását az első kettő, illetve a harmadik egységben! A felfutási és a lecsengési idő mindig nulla, a kitartási szint pedig mindig maximális, azaz 15. Az elengedési idő azonban az első két esetben, az önmagukban szóló hangoknál 11,10,10 (46., 150. sor), míg az akkordoknál 9,9,9-re csökken.

```
5 REM AL DI MEOLA & CSIKOS ZS.
6 PRINT"♥"
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1),A(32),B(32),C(32)
30 FORI=37TO62
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
45 POKES+24,15
46 POKES+6,251:POKES+13,250:POKES+20,250
50 FORI=1TO13:READX:A(I)=X+37:NEXT
55 FORI=1TO13:READX:B(I)=S+7*X:NEXT
60 FORI=1TO13:READC(I):NEXT
65 FORI=1TO8:READX:D(I)=X+37:NEXT
70 FORI=1TO8:READE(I):NEXT
75 FORI=1TO3:READX:F(I)=X+37:NEXT
100 FORL=1TO3
102 FORJ=1TO2
104 FORI=1TO 13
106 A=A(I):B=B(I):K=C(I)
108 POKEB+1,HB(A,1):POKEB,HB(A,0)
110 POKEB+4,33
112 FORT=1TOK:NEXT T
114 POKEB+4,32
116 NEXTI
118 NEXTJ
```

```
120 FORJ=1TO2:FORI=1TO8
122 A=D(I):B=B(I):K=E(I)
124 POKEB+1,HB(A,1):POKEB,HB(A,0):POKEB+4,33
128 FORT=1TOK:NEXT T
130 POKEB+4,32
132 NEXTI
138 POKES+6,249:POKES+13,249:POKES+20,249
139 FORI=1TO3
140 A=F(1):B=F(2):C=F(3):K=10
141 POKES+1,HB(A,1):POKES,HB(A,0)
142 POKES+8,HB(B,1):POKES+7,HB(B,0)
143 POKES+15,HB(C,1):POKES+14,HB(C,0)
144 POKES+4,33:POKES+11,33:POKES+18,33
145 FORT=1TOK:NEXT T
146 POKES+4,32:POKES+11,32:POKES+18,32
147 FORT=1TO10:NEXT
148 NEXTI
149 FORT=1TO70:NEXT
150 POKES+6,251:POKES+13,250:POKES+20,250
151 NEXTJ
152 FORT=1TO100:NEXT
153 NEXTL
300 DATA 20,21,25,18,21,25, 16,21,25,16,18,21,25
310 DATA 0,1,2,0,1,2, 0,1,2,0,0,1,2
320 DATA150,70,70,70,70,70, 150,70,70,90,30,70,80
330 DATA 8,15,16,20,16,15,8,1
331 DATA 70,70,70,70,90,150,100,300, 16,20,25
```

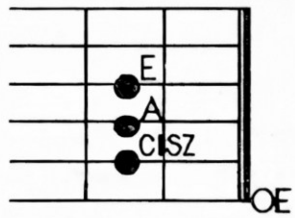
9. *példa*

Ez a program bizonyos fokig a véletlen szülötte. Eredetileg csak azt akartam kipróbálni, hogy mi történik, ha egy akkord hangjait gyors egymásutánban szólaltatom meg. Kíváncsi voltam, hogy a BASIC elég gyors-e ahhoz, hogy ily módon együtt halljam az akkordot. Nos, a válasz az, hogy a BASIC nem elég gyors, jól érzékelhető a különbség az egyes hangok között. A hatás azonban igen érdekes volt, ami - mint utólag kiderült -, részben a DATA sorok félregépelésének volt köszönhető. Így azután meghagytam olyannak, amilyenre sikerült, s kitaláltam hozzá egy basszus alapot, majd azt akkordokká bővítettem. A dallam ezeken az akkordokon lépdel fölfelé:



35. ábra

Mint már említettem, a dúr hármashangzat összetevői a prím, a nagyterc és a kvint, vagy közérthetőbben (szolmizálva) a dó, mi, szó hangok. Pl. az E-dúr akkordban E=dó, következésképp GISZ=mi, és H=szó, tehát az E-dúr akkord csak az E, GISZ és H hangokból állhat. Hasonlóan meghatározható a G-dúr és az A-dúr akkord is. Természetesen ezeket az akkordokat másként is játszhatjuk. Pl. a következő ábra is egy A-dúr akkordot mutat:



36. ábra

A dúr akkordokkal kapcsolatos kis kitérő után nézzük meg a 161-192 sorokat. Remélhetőleg most már világos e programrész feladata.

A 200-as és a 202-es sornak köszönhetően az akkord utolsó három hangjáról egy 8 hangnyi (16 félhangnyi) visszacsúszás következik. Végül, hogy ne maradjon üres a hangzás, a basszus alapra kitaláltam egy kíséretet, melynek adatait a D tömbben helyeztem el. Mivel a D tömb lényegesen több elemet tartalmaz, mint a basszus hangjait meghatározó B tömb, ezért gondoskodni kellett a basszus folyamatos ismétlődéséről is. Ezt úgy tudjuk elérni, hogy a B futóindexének egy F segédváltozót használunk, melynek értéke 16 után ismét 1 lesz. (206-216-os sor)

A 214-es sor szerepe a ritmuskiegyenlítés. Ennek hiányában ritmuskiesések fordulnának elő, mert a 212-es sorról hol gyorsabban, hol lassabban ugrana tovább a program. (Kipróbálhatjuk!)

```
5 REM CSIKOS ZS.
6 PRINT"□"
7 POKE53280,0:POKE53281,0
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1),A(16),B(16),A1(16),B1(16)
25 DIM C(16),C1(16),D(208)
30 FORI=1TO95
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
50 FORI=1TO5:READX:A(I)=X+40:A1(I)=A(I)+3:NEXT
51 FORI=1TO16:READX:C(I)=X+40:C1(I)=C(I)+3:NEXT
52 FORI=1TO16:READX:B(I)=X+35:B1(I)=B(I)+3:NEXT
53 FORI=1TO208:READX:D(I)=X+50:NEXT
60 POKES+24,15:POKES+6,240:POKES+13,251:POKES+20,255
70 H1=32:H2=32:H3=32
100 POKES+6,240:POKES+13,240
105 FORJ=1TO4
106 FORI=1TO16:K=A(I):L=A(16-I):M=C(I):GOSUB1000:NEXTI,J
110 FORJ=1TO2
111 FORI=1TO16:K=A1(I):L=A1(16-I):M=C1(I):GOSUB1000:NEXTI,J
115 FORJ=1TO2
116 FORI=1TO16:K=A(I):L=A(16-I):M=C(I):GOSUB1000:NEXTI,J
120 POKES+6,239:POKES+13,239:POKES+20,251
```

```
121 I1=0
122 FORJ=1TO6
124 FORI=1TO16:K=A(I):L=A(16-I):IFF-3THENM=B(I1):F=0
126 GOSUB1000:F=F+1:I1=I1+1:NEXTI
128 I1=0:NEXTJ
130 FORI=1TO16:K=A1(I):L=A(16-I):IFF-2THENM=B1(I1):F=0
132 GOSUB1000:F=F+1:I1=I1+1:NEXTI
140 POKES+6,247:H2=0:H3=0
141 FORJ=1TO3
142 FORI=1TO16:K=B(I):N=90:GOSUB1000:NEXTI,J
150 POKES+13,247:H2=32
151 FORJ=1TO3
152 FORI=1TO16:K=B(I):L=B(I)+7:N=70:GOSUB1000:NEXTI,J
160 POKES+20,247:H3=32
161 FORJ=1TO3
162 FORI=1TO16:K=B(I):L=K+7:M=L+5:N=70:GOSUB1000:NEXTI,J
171 FORJ=1TO2
172 FORI=1TO16:K=B(I)+7:L=K+5:M=L+4:N=70:GOSUB1000:NEXTI,J
181 FORJ=1TO2
182 FORI=1TO16:K=B(I)+12:L=K+4:M=L+3:N=70:GOSUB1000:NEXTI,J
191 FORJ=1TO2
192 FORI=1TO16:K=B(I)+16:L=K+3:M=L+5:N=70:GOSUB1000:NEXTI,J
200 FORI=16TO0STEP-1
202 K=B(I)+I:L=K+3:M=L+5:N=30:GOSUB1000:NEXT
205 FORI=1TO16:K=B(I):L=K:M=K:N=70:GOSUB1000:NEXT
206 F=1:POKES+20,235:POKES+6,250:POKES+13,250
210 FORI=1TO208:J=TI:K=B(F):L=K:M=D(I):N=30:GOSUB1000
212 F=F+1:IFF=17THENF=1
214 IFTI-J<11THEN 214
216 NEXT
217 GOTO205
300 DATA 0,7,12,15,19, 0,7,12,15,19, 0,7,12,15,19
304 DATA 0,0,0,0,0,0, 0,0,0,0,0,0,0,0
306 DATA 3,3,3,3,5,5,5,-35, 2,2,2,2,2,2,2,2
322 DATA 9,9,9,9,11,11,11,-50
324 DATA 11,-50,11,-50,9,-50,11,-50
326 DATA 12,-50,12,-50,11,-50,9,-50
328 DATA 11,-50,11,-50,9,-50,11,-50
330 DATA 12,-50,12,-50,11,-50,9,-50
332 DATA 11,9,7,6,4,2,4,6, 7,-50,6,-50,4,-50,-50,-50
336 DATA 11,9,7,6,4,2,4,6,  4,-50,2,-50,-3,-50,-50,-50
340 DATA 11,9,7,6,4,2,4,6, 7,9,11,-50,9,-50,-50,-50
344 DATA 11,9,7,6,4,2,4,6, 4,2,1,-50,2,-50,-50,-50
348 DATA 4,6,4,6,4,6,4,6, -7,1,2,-7,1,2,-7,-50
354 DATA 4,6,4,6,4,6,4,6, -7,1,2,-7,1,2,-7,-50
358 DATA -3,4,6,-3,4,6,-3,-50,6,-50,-50,-50,4,-50,6,-50
362 DATA -3,4,6,-3,4,6,-3,-50,4,-50,-50,-50,2,-50,-50,-50
366 DATA -3,4,6,-3,4,6,-3,-50,6,-50,-50,-50,4,-50,6,-50
370 DATA -3,4,6,-3,4,6,-3,-50,4,-50,-50,-50,2,-50,-50,0
900 END
1000 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006 IFKTHENPOKES+4,H1+1
1008 IFLTHENPOKES+11,H2+1
1010 IFMTHENPOKES+18,H3+1
1012 FORT=1TON:NEXT
1014 IFKTHENPOKES+4,H1
1016 IFLTHENPOKES+11,H2
1018 IFMTHENPOKES+18,H3
1020 RETURN
```

10. példa
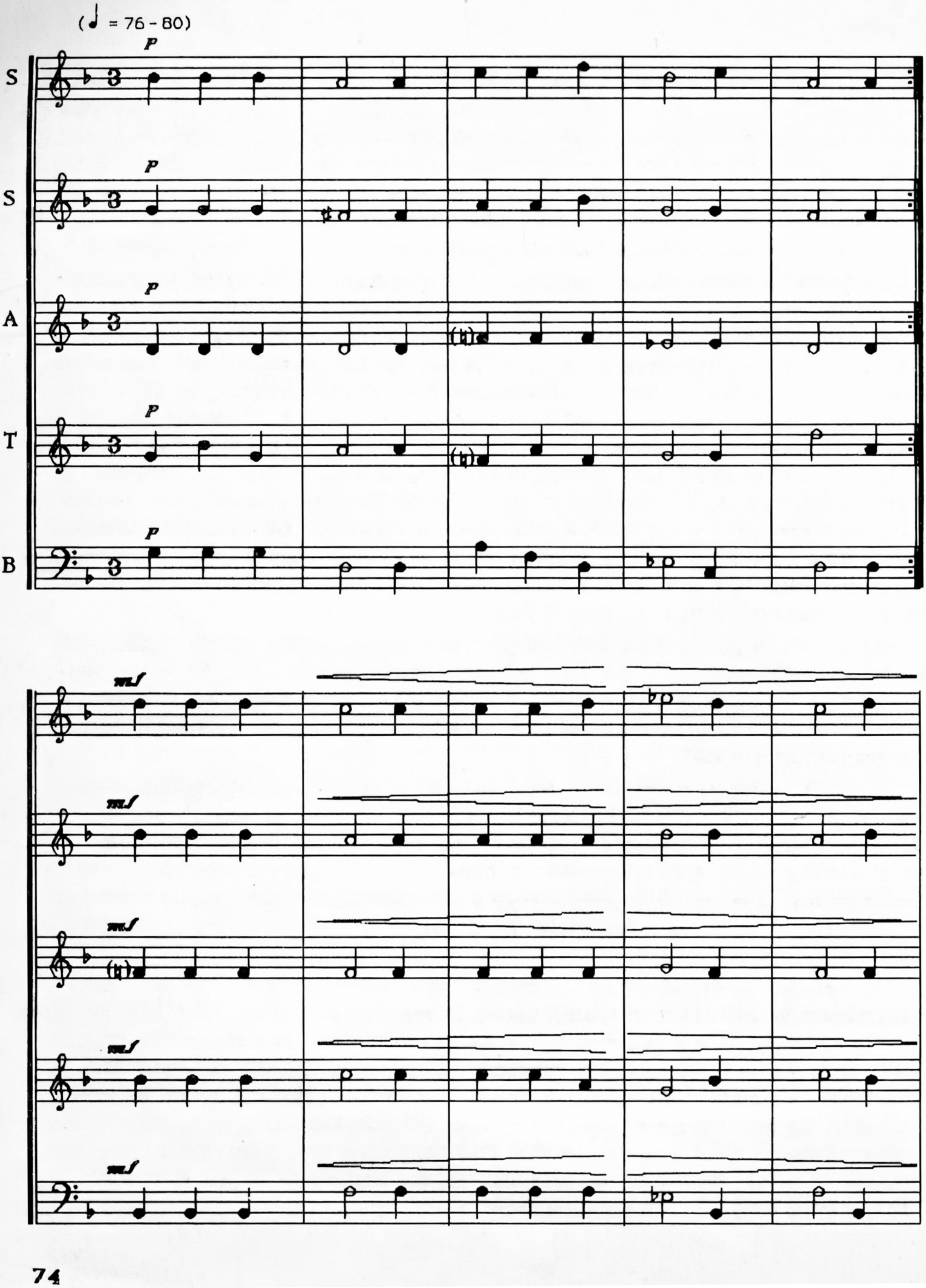
A fejezet utolsó példája Hans Leo Hassler (1564-1612) Őszi fák című ötszólamú darabjának, pontosabban e mű első felének két számítógépes változata. Nyilvánvaló, hogy hazánkban nemigen van olyan ember, akinek két Commodore gépe van otthon, azonban szakkörökben, klubokban, iskolákban valószínűleg találhatnak időnként két szabad C-64-est. (Ritka azonban az olyan pillanat, amikor senkit sem zavarunk zenélésünkkel, főleg az ilyen programok elkészítésének kezdeti szakaszában, amikor sokszor kell ugyanazt a dallamot kisebb módosításokkal újra és újra meghallgatunk.)

Technikailag nincs nehezebb dolgunk, mint az egygépes dalok esetében, csupán a két gép egyidejű indításáról kell gondoskodnunk a dallamjátszás kezdetekor. Ezt a 68, 69-es sorok biztosítják, arra kell csak ügyelni, hogy a gépeken egyszerre nyomjunk le egy-egy billentyűt (lehetőleg NE a RUN/STOP legyen az!), miután a két képernyőn megjelent az OK felirat. Arra is vigyáznunk kell, nehogy valamelyik gép játéka gyorsabb legyen a másikénál. Ezt a 70. sorban található N paraméter megfelelő megválasztásával tudjuk elérni. A listából láthatjuk, hogy a bal csatornán az N értéke 300, míg a jobbon 340. Ennek igen egyszerű a magyarázata. Először természetesen az utóbbi is 300 volt, de a programok futásakor a jobb csatorna gyorsabb volt a balnál, így a jobb csatornához tartozó N értékét növelni kellett. Az Olvasónak is bizonyára változtatnia kell majd ezen a számon, mert a C-64-es gépek nem egyformák. Háromszor futtattam le a programot különböző gépeken, és mindhárom esetben más és más értékeket kellett az N változóba írni.
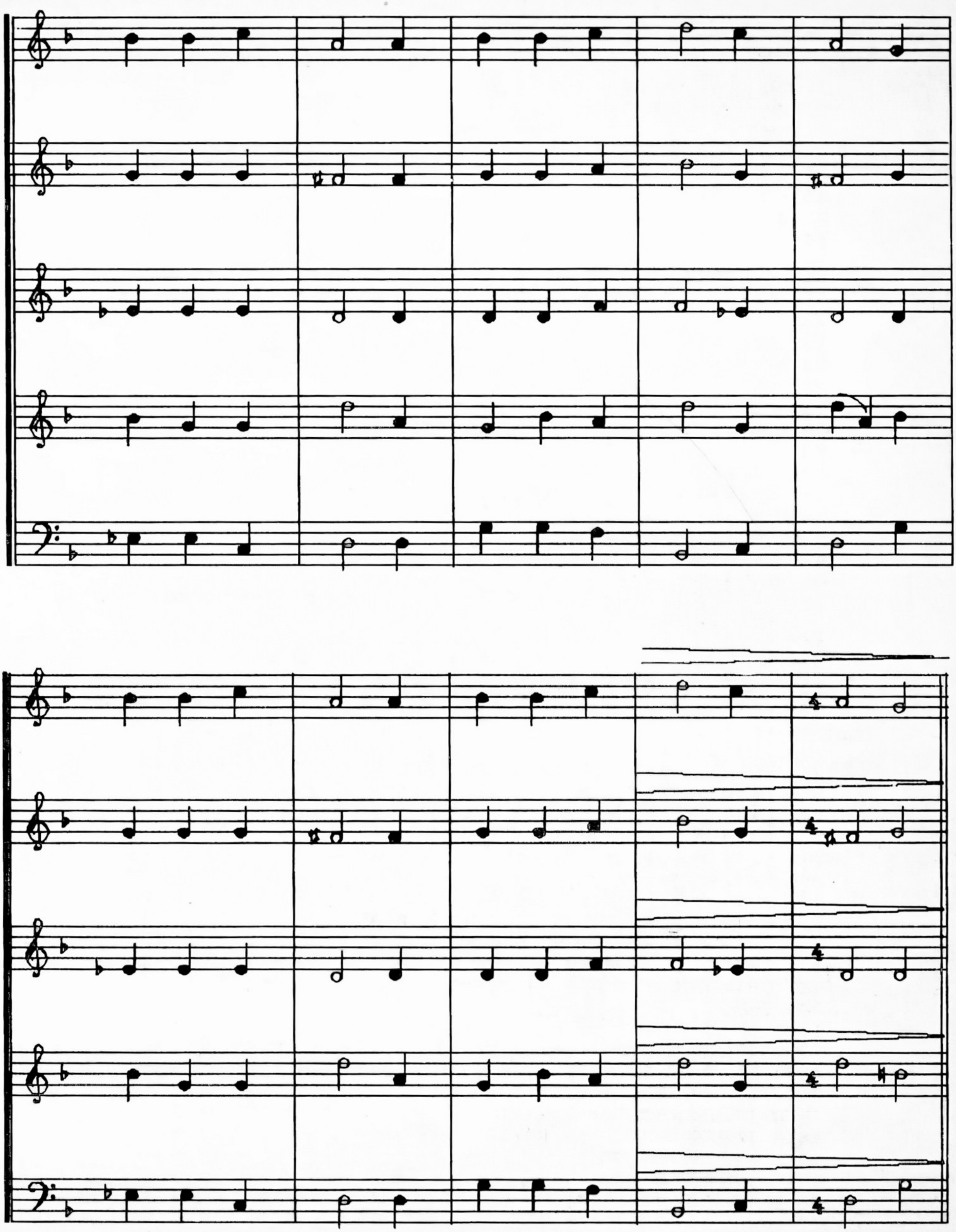
Az N változó persze csak azért tudja a programok futását összehangolni, mert egyforma azok szerkezete. A tenor kétcsatornás megszólaltatása is kettős célú: egyrészt az öt szólamból így hat lett - hiszen a két tenor hangszíne különböző -, másrészt a programok futását is könnyebben lehet egyeztetni, ha a két számítógépen ugyanannyi szólam szól.

A DATA sorokat megfigyelve feltűnhet, hogy most nem használunk negatív számot. Az eddigiekben a DATA sorokban szereplő -1-esek jelentették azt, hogy a hangmagasságokat meghatározó tömbök megfelelő eleme nulla lesz. Ebben a programban a -1 szerepét átveszi a 0, így minden esetben eggyel kevesebb billentyűt kell leütnünk. Ez persze azt is jelenti, hogy a szólamok legmélyebb hangját most nem nulla, hanem 1 jelöli, és a beolvasást is egy kicsit módosítani kellett. (lásd 50-55 sorok)

Az eredeti kottától ismét eltértünk egy kissé. A két szoprán szólam hangmagassága egy oktávval feljebb került. Ez zeneileg semmi változást nem jelent, viszont megszólaláskor e szólamok sokkal jobban hallhatóak. Ez elsősorban a szoprán 1 esetében lényeges, mert ennek hangszínét a háromszög hullám határozza meg. A különböző dinamikai előírásokat sem vettük figyelembe, hogy a program ne legyen túl hosszú. Ha azonban igazán klassz programot akarunk készíteni, akkor ezekről sem szabad elfeledkeznünk, s nagyon hatásos, ha a hallgatóság érzi, hogy egyszer az egyik, másszor a másik csatorna szól dinamikusabban. Esetleg az egyes szólamokat is váltakoztatva hangsúlyozhatjuk a darabnak megfelelően.


(♩ = 76 - 80)
S
S
A
T
B
p
mf

Dallamok programozása



```
1 REM HASSLER:OSZI FAK
2 REM  /BALCSATORNA/
3 REM
5 PRINT"♥"
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1),S1(76),A(76),T(76)
30 FORI=50TO76
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
50 FORI=1TO76:READX:IFX THENS1(I)=X+65
51 NEXT
52 FORI=1TO76:READX:IFX THEN A(I)=X+50
53 NEXT
54 FORI=1TO76:READX:IFX THEN T(I)=X+53
55 NEXT
60 POKES+24,15
62 POKES+6,252:POKES+12,11:POKES+13,139:POKES+20,156
64 POKES+9,130:POKES+17,8
66 H1=16:H2=64:H3=64
68 PRINT"OK"
69 GETX$:IFX$=""THEN 69
70 FORI=1TO76:K=S1(I):L=A(I):M=T(I):N=300:GOSUB1000:NEXT
300 REM ** SOPRAN 1 **
301 DATA 6,6,6, 5,0,5, 8,8,10, 6,0,8, 5,0,5
302 DATA 6,6,6, 5,0,5, 8,8,10, 6,0,8, 5,0,5
303 DATA 10,10,10, 8,0,8, 8,8,10, 11,0,10, 8,0,10
304 DATA 6,6,6, 5,0,5, 6,6,8, 10,0,8, 5,0,3
305 DATA 6,6,8, 5,0,5, 6,6,8, 10,0,8, 5,0,3,0
310 REM ** ALT **
311 DATA 1,1,1, 1,0,1, 4,4,4, 2,0,2, 1,0,1
312 DATA 1,1,1, 1,0,1, 4,4,4, 2,0,2, 1,0,1
313 DATA 4,4,4, 4,0,4, 4,4,4, 6,0,4, 4,0,4
314 DATA 2,2,2, 1,0,1, 1,1,4, 4,0,2, 1,0,1
315 DATA 2,2,2, 1,0,1, 1,1,4, 4,0,2, 1,0,1,0
320 REM ** TENOR **
321 DATA 3,6,3, 5,0,5, 1,5,1, 3,0,3, 10,0,5
322 DATA 3,6,3, 5,0,5, 1,5,1, 3,0,3, 10,0,5
323 DATA 6,6,6, 8,0,8, 8,8,5, 3,0,6, 8,0,6
324 DATA 6,3,3, 10,0,5, 3,6,5, 10,0,3, 10,5,6
325 DATA 6,3,3, 10,0,5, 3,6,5, 10,0,3, 10,0,7,0
900 END
1000 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006 IFKTHENPOKES+4,H1+1
1008 IFLTHENPOKES+11,H2+1
1010 IFMTHENPOKES+18,H3+1
1012 FORT=1TON:NEXT
1014 IFKTHENPOKES+4,H1
1016 IFLTHENPOKES+11,H2
1018 IFMTHENPOKES+18,H3
1020 RETURN
```

```
1 REM HASSLER:OSZI FAK
2 REM  /JOBBCSATORNA/
3 REM
4 PRINT"♥"
10 S=54272
15 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1),S2(76),B(76),T(76)
30 FORI=34TO71
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
50 FORI=1TO76:READX:IFX THENS2(I)=X+62
51 NEXT
52 FORI=1TO76:READX:IFX THEN B(I)=X+34
53 NEXT
54 FORI=1TO76:READX:IFX THEN T(I)=X+53
55 NEXT
60 POKES+24,15
62 POKES+6,156:POKES+13,252:POKES+20,155
66 H1=32:H2=16:H3=32
68 PRINT"OK"
69 GETX$:IFX$=""THEN69
70 FORI=1TO76:K=S2(I):L=T(I):M=B(I):N=340:GOSUB1000:NEXT
300 REM ** SOPRAN 2 **
301 DATA 6,6,6, 5,0,5, 8,8,9, 6,0,6, 5,0,5
302 DATA 6,6,6, 5,0,5, 8,8,9, 6,0,6, 5,0,5
303 DATA 9,9,9, 8,0,8, 8,8,8, 9,0,9, 8,0,9
304 DATA 6,6,6, 5,0,5, 6,6,8, 9,0,6, 5,0,6
305 DATA 6,6,6, 5,0,5, 6,6,8, 9,0,6, 5,0,6,0
310 REM ** BASSZUS **
311 DATA 10,10,10, 5,0,5, 12,8,5, 6,0,3, 5,0,5
312 DATA 10,10,10, 5,0,5, 12,8,5, 6,0,3, 5,0,5
313 DATA 1,1,1, 8,0,8, 8,8,8, 6,0,1, 8,0,1
314 DATA 6,6,3, 5,0,5, 10,10,8, 1,0,3, 5,0,10
315 DATA 6,6,3, 5,0,5, 10,10,8, 1,0,3, 5,0,10,0
320 REM ** TENOR **
321 DATA 3,6,3, 5,0,5, 1,5,1, 3,0,3, 10,0,5
322 DATA 3,6,3, 5,0,5, 1,5,1, 3,0,3, 10,0,5
323 DATA 6,6,6, 8,0,8, 8,8,5, 3,0,6, 8,0,6
324 DATA 6,3,3, 10,0,5, 3,6,5, 10,0,3, 10,5,6
325 DATA 6,3,3, 10,0,5, 3,6,5, 10,0,3, 10,0,7,0
900 END
1000  IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002  IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004  IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006  IFKTHENPOKES+4,H1+1
1008  IFLTHENPOKES+11,H2+1
1010  IFMTHENPOKES+18,H3+1
1012  FORT=1TON:NEXT
1014  IFKTHENPOKES+4,H1
1016  IFLTHENPOKES+11,H2
1018  IFMTHENPOKES+18,H3
1020  RETURN
```

Nem maradt más hátra, mint hogy összefoglaljuk a fejezetben elmondottakat.

A hangok megszólalásához szükséges paraméterek beolvasása történhet:

- a zenélő rész közben
- a zenélő rész előtt elhelyezett tömbökből.

A hangok frekvenciáját, magasságát a következő módokon tudjuk meghatározni:

- az 1. táblázat alapján, a P paraméterek beolvasásával
- az 1. táblázat alapján, az F és A paraméterek beolvasásával
- az 5. példaprogram használata esetén a hangmagasságok egymáshoz viszonyított helyzetével és a beolvasás utáni pontosítással.

A hang kitartásának idejét is többféleképpen szabályozhatjuk:

- üres ciklussal (az 5. program használata esetén az N változóval)
- a C-64 belső időmérőjével, a TI és TI$ használatával
- egyéb módon, pl. a kapu-bit zárása és nyitása közötti grafikával (erről részletesebben az 5. fejezetben lesz szó).

Fontosnak tartom megjegyezni, hogy a fejezet célja nem egyetlen, minden zeneszám programozására alkalmas eljárás bemutatása volt. A különböző programok, programozási módszerek megismerése éppen azt szolgálja, hogy minden esetben ki tudjuk választani a legmegfelelőbbet, illetve, hogy szükség esetén képesek legyünk további programozási trükkök kidolgozására.

Aki idáig eljutott a könyv olvasásában, már mindent tud ahhoz, hogy egy tetszőleges dallamot megszólaltasson a C-64-en. Azoknak, akik ebbe szívesen belevágnának, csak még nincs erre alkalmas kottájuk, a [10] és [11] kötetek tanulmányozását javaslom.

Szintetizátor

A fejezetben egy BASIC-ben írt szintetizátor-programot fogunk alaposan áttanulmányozni. Sajnos a program segítségével a gyors játék nem oldható meg tökéletesen. Mint látni fogjuk, a gyorsaságnak itt nem is a BASIC, hanem elsősorban a billentyűzet szab határt. A C-64 billentyűzete messze nem hasonlítható össze egy valódi szintetizátor klaviatúrájával. Ennek ellenére több okból igen hasznos lehet egy ilyen program megírása.

- A program segítségével alaposabban megismerhetjük a SID chip által biztosított lehetőségeket.
- Az egyes részeket fokozatosan átírhatjuk gépi kódú rutinokká, bár ebben az esetben egy-két programozási trükköt nem árt még becsempészni.
- Programozási ismereteinket is bővíti a program megismerése.

Először ismerkedjünk meg a program vázával!

A megnyomott billentyűk lekérdezése többféleképpen történhet. Az egyik lehetőséget a billentyűzet-puffer biztosítja. A GET utasítás segítségével megtudhatjuk, hogy melyik a legrégebben megnyomott billentyű, melynek adatát még nem dolgozta fel a gép. Hogy a programunkban nem ezt a lekérdezésfajtát használjuk, annak két oka van. Az első, hogy a C-64-en nem célszerű gyakran használni a szöveges változókat. A második, hogy fáziseltolódás lehet a billentyűk leütése és a feldolgozás között, - illetve ez a fáziseltolódás fokozatosan nőhet. Így elméletileg az is elképzelhető, hogy valamelyik lenyomott billentyű adata elvész (a billentyűzetpuffer csak az utolsó 20 leütött billentyűt tudja megjegyezni, ami könnyen kalamajkát okozhat).

A billentyűzet lekérdezésére adódó másik lehetőség: a memória 197. byte-ján levő érték megvizsgálása. Ez az érték ugyanis azt mutatja meg, hogy éppen milyen billentyű van lenyomva. A következő táblázat foglalja össze, hogy az egyes billentyűk lenyomásakor milyen értéke lesz a 197. byte-nak. (A C-64 203. byte-ja is ugyanígy működik)

←	1	2	3	4	5	6	7	8	9	0	+	-	£	INST DEL	CLR HOME
57	56	59	8	11	16	19	24	27	32	35	40	43	48	51	0

Q	W	E	R	T	Y	U	I	O	P	@	*	↑
62	9	14	17	22	25	30	33	38	41	46	49	54

A	S	D	F	G	H	J	K	L	:	;	=	RETURN
10	13	18	21	26	29	34	37	42	45	50	53	1

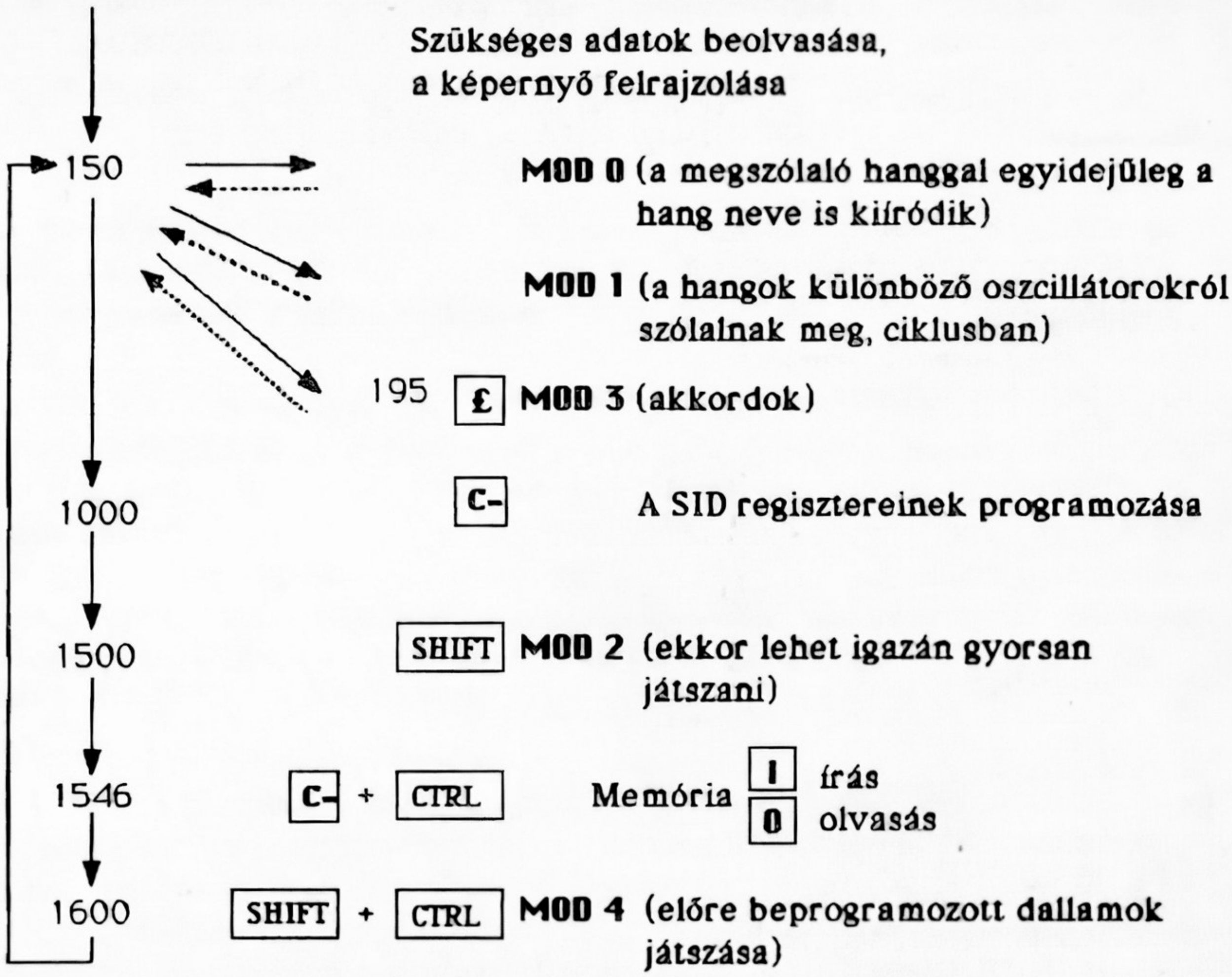
Z	X	C	V	B	N	M	,	.	/	CRSR ⇕	CRSR ⇔
12	23	20	31	28	39	36	47	44	55	7	2

SPACE BILLENTYŰ: 60

Ha nincsen billentyű megnyomva, a 197. byte értéke 64. Ezen byte értékéből azonban még nem derül ki, hogy pl.a SHIFT vagy a CTRL billentyűvel együtt nyomtuk-e meg a billentyűt, vagy csak önmagában. Ennek megkülönböztetésére szolgál a memória 653. byte-ja. Ennek értéke a következők szerint alakul:

0, ha a SHIFT , C= és a CTRL billentyűk közül egyik sincs lenyomva;
1, ha a SHIFT billentyű van lenyomva;
2, ha a C= billentyű van lenyomva;
3, ha a SHIFT és a C= billentyűk egyszerre vannak lenyomva;
4, ha a CTRL billentyű van lenyomva;
5, ha a SHIFT és a CTRL billentyűk egyszerre vannak lenyomva;
6, ha a C= és a CTRL billentyűk egyszerre vannak lenyomva;
7, ha a SHIFT, a C=, és a CTRL billentyűk is le vannak nyomva.

Ezek alapján a program váza a következőképpen néz ki:



Az egyes MOD-okból a SPACE billentyű lenyomásával tudunk kilépni, ha az egyes váltóbillentyűket (pl. MOD2 esetében a SHIFT billentyűt) előzetesen felengedtük. Hasonlóan a memória írásának befejezését is a SPACE billentyű megnyomása jelenti. Kezdjük el programunk alapos vizsgálatát! Ennek révén szintetizátorunk használatára is fény derül.*

---

* A program begépelését célszerű az adatokkal, azaz a DATA sorokkal kezdeni. Ezek listája a fejezet végén található.

```
1 CLR:PRINT"□"
2 POKE53280,0 :POKE53281,0:POKE646,1
3 DIMHANG(65),NEV$(23),HB(96,1)
4 DIM R1(44),P1(44),R2(79),P2(79),Q2(79)
5 DIM P3(95),P4(96)
7 DIM ID(200),VE(200),AH(200),R6(47),P6(47)
20 PRINT"□■■■"TAB(12)"SZINTETIZATOR"
21 PRINTTAB(5)"■■ ■ | ■ ■ | ■ ■ ■ | ■ ■ | ■ ■ "
22 PRINTTAB(5)  " ■ | ■ ■ | ■ ■ ■ | ■ ■ | ■ ■ "
23 PRINTTAB(5)  " ■ | ■ ■ | ■ ■ ■ | ■ ■ | ■ ■ "
24 PRINTTAB(5)  " ■ | | | | | | | | | | | | "
25 PRINTTAB(5)  " ■|A|S|D|F|G|H|J|K|L|:|;|=| "
30 FORI=1TO3:OKT(I)=5:NEXT:E=1
31 FORI=1TO64:READHANG(I):NEXT I
32 FORI=0TO11:READNEV$(I):NEV$(I+12)=NEV$(I):NEXT
33 FORI=1TO96
34 FR=248.0858*2↑(I/12)
35 HB(I,0)=INT(FR/256):HB(I,1)=INT(FR-256*HB(I,0))
36 NEXT
40 FORT=0TO44:READR1(T):NEXT
41 FORT=0TO44:READP1(T):NEXT
43 FORT=0TO79:READR2(T):NEXT
44 FORT=0TO79:READP2(T):NEXT
45 FORT=0TO79:READQ2(T):NEXT
46 FORT=0TO95:READP3(T):NEXT
48 FORT=0TO95:READP4(T):NEXT
49 POKE646,8
58 S=54272:FORL=0TO24:POKES+L,0:NEXT
60 V1=1:V2=1:V3=1
100 PRINT"□■VOL +       "
101 PRINT"OKT:          OSC1         OSC2         OSC3"
102 PRINT"INF:"
103 PRINT"ATTACK   A"
104 PRINT"DECAY    S"
105 PRINT"SUSTAIN  D"
106 PRINT"RELEASE  F"
107 PRINT"WAVE         ◆▒┴◥ S●X    ◆▒┴◥ S●X    ◆▒┴◥ S●X"
108 PRINT"             ZXCV BNM"
109 PRINT"■PW L        Q"
110 PRINT"PW H       W"
111 PRINT"■KI:     ?"
112 PRINT"SZ:   >"
113 PRINT"┌──────────────────────────▌────────────────────"
114 PRINT"SZURES   FR L   E       ▌ ROCK         MOD:"
115 PRINT"1 2 3 A FR H   R       ▌1 2 3 4 5 6 7 8 9"
116 PRINT"[ ] =                  ▌"
117 PRINT"ALUL   T    REZON O    ▌"
118 PRINT"SAV    Y     ┌────────    "
119 PRINT"FELUL  U     ▌ MEMORY"
120 PRINT"3.LE   I     ▌ ■I■R ■■■LV"
```

A 2-es sorban a képernyő háttér- és keretszínét állítjuk be feketére, a kurzor színét pedig fehérre. A dimenzionált tömbök jelentésére később térünk vissza. A 20-25-ös sorok hatására a szintetizátor billentyűzetének rajza jelenik meg a képernyőn, mutatva azt is, hogy a C-64 mely billentyűin tudunk majd játszani. Eszerint az A, S, D, F, G, H, J, K, L, :, ;, =, RETURN billentyűk felelnek meg az egész hangoknak – mégpedig a c hangtól indulva –, a Q, W, E, T, Y, U, O, P, *, ↑ pedig a megfelelő félhangoknak. Az R, I és a @ billentyűk nem véletlenül maradtak ki, hanem azért,

mert ezeken a helyeken nincs félhang. A Q billentyű esetében választani kellett, hogy vagy az előbbiekhez hasonlóan kimarad, vagy pedig a h hangot fogja reprezentálni. (A h és a c hang között ugyanis szintén nincs félhang.) A könnyebb kezelhetőség kedvéért az utóbbi megoldást választottam.

A 33-36 sorokban töltjük fel a HB tömböt. Ez a tömb ugyanolyan szerepet fog játszani, mint az előző fejezetben. A HB (1,0) és a HB (1,1) a H-0 hangot határozzák meg.

A 49-es sorban ismét a kurzor színét változtatjuk. A 100-120 sorokban töröljük az előbbi képernyőt és egy újat rajzolunk fel. Ez egyben azt is jelenti, hogy a szintetizátorunkhoz szükséges adatok beolvasása megtörtént, elkezdhetjük majd a zenélést.

A 100-120. sorok írják ki a képernyőre, hogy az egyes paramétereket melyik billentyűvel tudjuk állítani. Fussuk át gyorsan, hogy mi mit jelent a képernyőn! Mivel három hanggenerátor van, tudnunk kell, hogy az egyes paramétereket melyik hanghoz állítjuk be, illetve, hogy hogyan válthatunk át az egyik oszcillátorról (OSC) egy másikra. Nos, egyszerűen a C= és az 1, 2, ill. 3 billentyűk valamelyikének lenyomásával tudjuk kiválasztani a megfelelő oszcillátort. A paraméterek beállítására csak ezután kerülhet sor. A választott oszcillátort - vagyis amelynek paramétereit éppen módosíthatjuk - a megfelelő OSC jel alatti kis karika (•) jelöli.

A további rövidítések jelentése:

- a VOL a volument, azaz a hangerőt jelzi, az utána következő "+" pedig azt mutatja, hogy ezzel a billentyűvel lehet a hangerőt szabályozni. Ha ismét megnézzük a program vázának ismertetését, észrevehetjük, hogy a + billentyű - és a szintetizátorunk további paramétereit irányító billentyűk is - csak akkor változtatják a megfelelő paramétert, ha azokkal egyidőben a C= billentyűt is lenyomjuk. A hangerő maximuma 15.

- az OKT a hang oktávját jelenti, ez a kurzor-mozgató billentyűkkel változtatható felfelé, ill. lefelé (az oktáv maximális értéke 6, minimális értéke pedig 0). Sajnos az ezeknek megfelelő jelek hely hiányában nem kerülhettek fel a képernyőre, tehát meg kell jegyeznünk őket. Az ATTACK, DECAY, SUSTAIN és RELEASE a megszólaló hang dinamikáját szabályozó négy parraméter. Ezeket rendre az A, S, D ill. F billentyűkkel tudjuk beállítani. Maximális értékük 15. A WAVE szó magyarul hullámot jelent, a vele egy sorban levő jelek ezzel kapcsolatosak.

♦	fehér zaj
▒	négyszöghullám
⊥	fűrészfoghullám
◥	háromszöghullám
S	szinkronizáció
●	körmoduláció
×	a hang leválasztása az audio kimenetről

A hullám tulajdonságait rendre a Z,X,C,V,B,N, illetve M billentyűk megnyomásával tudjuk kiválasztani.

- a PW L (Pulse width low) az impulzus szélesség alsó byte-ja; a PW H (Pulse width high) az impulzus szélesség felső byte-ja négyszöghullám esetén. Ezeket a Q, ill. W billentyűkkel változtathatjuk.

- KI a hang kitartását meghatározó időt jelenti, az SZ pedig a hang elengedésére biztosított időt, szünetet.

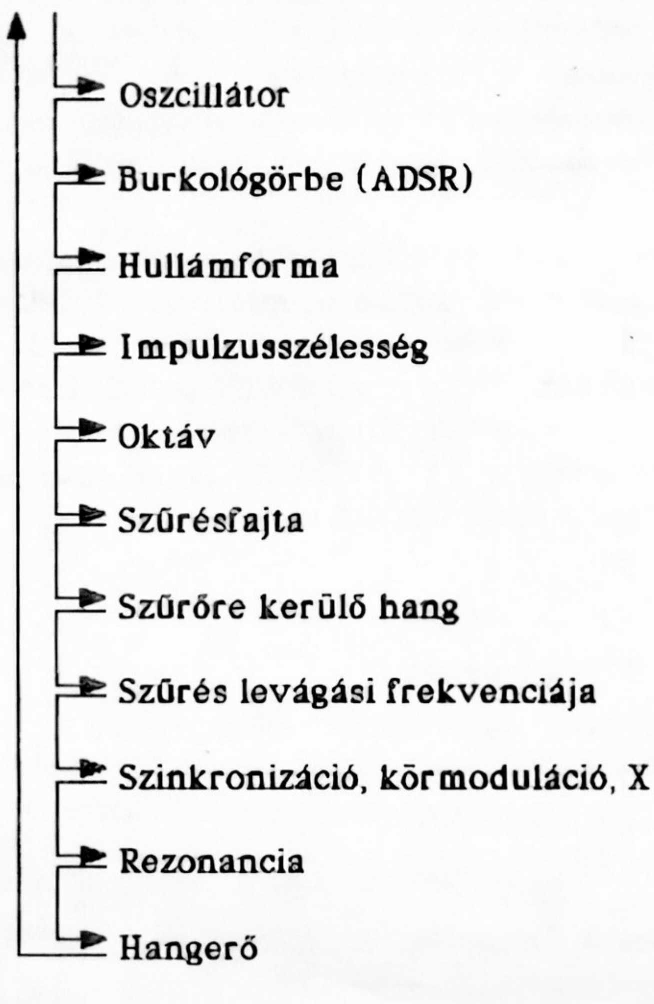
- a szűrést meghatározó paraméterek alakulását a képernyő bal alsó sarkában kísérhetjük figyelemmel. Attól függően, hogy az 1.,2. vagy 3. hangot, avagy az audio bemeneten megjelenő hangot akarjuk-e szűrni, a [,], = ill. a RETURN billentyűt kell megnyomni. (Az utóbbi billentyűt szintén nem lehetett a képernyőn jelölni, de könnyű megjegyezni, hiszen ez követi sorrendben az előző hármat.)

- a szűrés fajtáját, ill. a harmadik hang közvetlen leválasztását az audio kimenetről a T,Y,U ill. az I billentyűvel szabályozhatjuk, a levágási ferekvenciát pedig az E és R billentyűkkel (utóbbi határozza meg a felső byte-ot).

- a rezonancia az O billentyűvel állítható be.

Most már a SID chip által biztosított összes lehetőséget kezelni tudjuk a program segítségével. A következő lépés: megnézzük, miként valósul meg a programban a paraméterek változtatása.

A megnyomott billentyű lekérdezése aszerint történik, hogy mit akarunk vele változtatni.



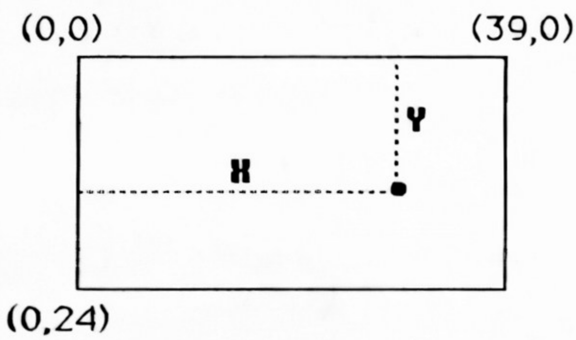
```
1000 IFPEEK(653)-2 THEN 1500
1005 IFPEEK(197)=64 THEN 1000
1010 M=PEEK(197)
1020 IF M=56THENH=1:H1=2:H2=3:GOTO2000
1021 IF M=59THENH=2:H1=1:H2=3:GOTO2000
1022 IF M= 8THENH=3:H1=1:H2=2:GOTO2000
1025 IFH=0THEN 150
```

Láthatjuk, hogy amíg a C= billentyűn kívül nem nyomunk le más billentyűt – azaz PEEK (653)=2 és PEEK (197) = 64 –, addig a program az 1000 és az 1005-ös sor között ugrál. Így lehetőség van arra, hogy egy véletlenül megnyomott C= billentyű hatására ne kelljen mindenképpen megváltoztatni valamelyik paramétert (ill. egy semleges billentyű lenyomásával fecsérelni az időt). Ha egy normál* billentyűt nyomunk le, az őt jellemző számot, a 197. byte értékét azonnal kimentjük egy változóba (M), nehogy elvesszen.

Ezután következik annak megvizsgálása, hogy vajon az 1, 2, ill. 3. billentyűk valamelyikét nyomtuk-e meg, vagy sem. Ha igen, akkor a H változóba tesszük a megfelelő számot, tehát ettől fogva H értéke fogja megmutatni, hogy melyik oszcillátor paramétereire vonatkoznak a változtatások. (H1,H2 változóknak csak a későbbiekben lesz szerepe.) A 2000-2010 utasítások hatására egy kis karika kerül a megfelelő OSC jel alá, míg a korábbi karika törlődik.

```
2000 POKE211,3+10*H1:POKE214,3:SYS58640:PRINT" "
2001 POKE211,3+10*H2:POKE214,3:SYS58640:PRINT" "
2002 POKE211,3+10*H:POKE214,3:SYS58640:PRINT"o"
2010 GOTO150
```

A képernyő egy adott pozíciójába - most, és a továbbiakban is - a következő módon írunk:



POKE 211,X: POKE 214,Y
SYS 58640: PRINT "●"

37. ábra

Ha még egyik oszcillátort sem választottuk ki, akkor az 1025-ös sor lehetetlenné teszi egyéb paraméterek beállítását.

A következő programrész a burkológörbe (ADSR) paramétereinek változtatását biztosítja.

---

* A továbbiakban így nevezzük a CTRL, SHIFT, C=, RUN/STOP és RESTORE billentyűkön kívüli billentyűket, azaz amelyek megnyomásakor a 197. byte értéke 64-től különböző lesz.

```
1050 IFM-10THEN 1060
1051 A(H)=A(H)+1-16*INT((A(H)+1)/16)
1052 K=S-2+7*H
1053 POKEK,D(H)+16*A(H)
1054 IF A(H)<10THENA=2+10*H:B=4:C=A(H):GOTO3010
1055 IF A(H)>=10THENA=1+10*H:B=4:C=A(H):GOTO3010
1060 IFM-13THEN 1070
1061 D(H)=D(H)+1-16*INT((D(H)+1)/16)
1062 K=S-2+7*H
1063 POKEK,16*A(H)+D(H)
1064 IF D(H)<10THENA=2+10*H:B=5:C=D(H):GOTO3010
1065 IF D(H)>=10THENA=1+10*H:B=5:C=D(H):GOTO3010
1070 IFM-18THEN 1080
1071 S(H)=S(H)+1-16*INT((S(H)+1)/16)
1072 K=S-1+7*H
1073 POKEK,R(H)+16*S(H)
1074 IF S(H)<10THENA=2+10*H:B=6:C=S(H):GOTO3010
1075 IF S(H)>=10THENA=1+10*H:B=6:C=S(H):GOTO3010
1080 IFM-21THEN 1090
1081 R(H)=R(H)+1-16*INT((R(H)+1)/16)
1082 K=S-1+7*H
1083 POKEK,16*S(H)+R(H)
1084 IF R(H)<10THENA=2+10*H:B=7:C=R(H):GOTO3010
1085 IF R(H)>=10THENA=1+10*H:B=7:C=R(H):GOTO3010
```

Mivel a SID első 24 regisztere nem olvasható, ezért külön tömbökben kell megjegyezni a paraméterek pillanatnyi értékét ahhoz,hogy ezen legközelebb módosítani tudjunk. Az egyes oszcillátorokhoz tartozó ADSR értékeket rendre az A(H), S(H), és R(H) tömbökben tároljuk, ahol H=1,2,3. Az ADSR négy szakaszában lényegében ugyanazt csinálja a program, ezért most csak az első esetet fogjuk részletesebben átnézni.

Mint már volt róla szó, a felfutási időt (attack) az A billentyű lenyomásával tudjuk szabályozni. Ez azt jelenti, hogy az A billentyűt lenyomva tartva a felfutási idő folyamatosan nő 0-ról 15-re, ezután ismét 0 lesz, és újra növekedni kezd. Az 1051-es sor alapján:

$$A(H) = \begin{cases} A(H) + 1 & \text{ha } A(H) < 15 \\ \emptyset & \text{ha } A(H) = 15 \end{cases}$$

Az 1052-1053-as sorokban írjuk be a megfelelő regiszterbe az új értéket. Az 1054-1055-ös sorok a képernyőre írják a felfutási idő új értékét.

```
3010 POKE211,A:POKE214,B:SYS58640:PRINTC
3015 GOTO150
```

A hullámformát az alábbi sorok kezelik:

```
1089 REM** HULLAMFORMA **************
1090 IFM-12THEN 1092
1091 I=7:GOTO1100
1092 IFM-23THEN 1094
1093 I=6:GOTO1100
1094 IFM-20THEN 1096
1095 I=5:GOTO1100
1096 IFM-31THEN 1110
1097 I=4
1100 IFHF(H)AND2↑ITHEN 1105
1101 HM=HF(H)OR2↑I
1102 IF HM>128+15 THEN 1000
1103 HF(H)=HM
1104 A=10*H+6-I:B=10:C$="¯":GOTO3000
1105 HF(H)=HF(H)AND(255-2↑I)
1106 A=10*H+6-I:B=10:C$=" ":GOTO3000
```

A programrész feladatai:

1. ha megfelelő billentyűt nyomunk le, akkor az ahhoz tartozó hullámformát válassza ki az illető oszcillátor számára, illetve, ha előzőleg már ez a hullámforma volt kiválasztva, akkor szüntesse meg.
2. ne engedjen meg olyan kevert hullámformát, amelynek az egyik tagja fehér zaj.

Ezeket a következőképpen valósítja meg: A HF(H) /H=1,2,3/ tömbben tárolja az egyes kontroll regiszterek tartalmát. Ennek a 7., 6., 5., és 4. bitjei határozzák meg a hullámformákat. Az I tehát éppen a megfelelő bit sorszámával lesz egyenlő (1090-1097). Az 1100-as sor vizsgálja meg, hogy ez a bit magas-e vagy alacsony. Ha magas, akkor ezt a hullámformát ki kell kapcsolni a bit nullára állításával (1105), és ennek eredménye az 1106-os sor hatására a képernyőn is megjelenik. Amennyiben a kérdéses bit eddig nulla volt, akkor most magasra kell állítani (1101), feltéve, hogy nem alakul ki olyan kevert hullámforma, melynek egyik tagja fehér zaj (1102-1103). Ha a bit magasra állítása megtörtént, ezt szintén jelzi a program a képernyőn (1104).

```
3000 POKE211,A:POKE214,B:SYS58640:PRINTC$
3005 GOTO150
```

A hullámformával kapcsolatos további lehetőségek programozásának részletes áttanulmányozását az Olvasóra bízom. Négyszöghullám esetén az impulzusszélességét is be kell állítanunk. Erre, mint tudjuk, oszcillátoronként két-két byte áll rendelkezésre. Az alsó byte beállítását az 1108-1116, a felsőét a 1119-1126 sorok végzik.

```
1108 REM** IMP.SZELESSEG ***************
1110 IFM-62THEN 1120
1111 ISZ(H)=ISZ(H)+1-256*INT(ISZ(H)/255)
1112 IFISZ(H)THEN 1115
1113 POKE211,10*H+2:POKE214,11:SYS58640:PRINT" "
1114 POKE211,10*H+3:POKE214,11:SYS58640:PRINT" "
1115 POKES-5+7*H,ISZ(H)
1116 A=10*H:B=11:C=ISZ(H):GOTO3010
1119 REM**FELSO BYTE******
1120 IFM-9THEN 1130
1121 FB(H)=FB(H)+1-16*INT(FB(H)/15)
1122 IFFB(H)THEN 1125
1123 POKE211,10*H+2:POKE214,12:SYS58640:PRINT" "
1125 POKES-4+7*H,FB(H)
1126 A=10*H:B=12:C=FB(H):GOTO3010
```

M Feladat 15.

Milyen célt szolgálnak az 1112-1114 sorok? Hogyan lehetne egy sorral rövidíteni programunkat?

A következő lépés az oktáv kezelésének megoldása. A kurzort fel-le mozgató billentyű megnyomására (M=7) az oktávot eggyel feljebb kell emelni, ha annak értéke a maximális 6-nál kisebb, illetve a kurzort vízszintesen mozgató billentyű megnyomására (M=2) eggyel csökkenteni kell, ha értéke a minimális nullánál nagyobb (1130-1137).

```
1129 REM*** OKTAV ********
1130 IFM-7THEN 1135
1131 OKT(H)=OKT(H)+1-INT(OKT(H)/6)
1132 A=10*H-2:B=2:C=OKT(H):GOTO3010
1135 IFM-2THEN 1140
1136 OKT(H)=OKT(H)-1+INT(1/(OKT(H)+0.7))
1137 A=10*H-2:B=2:C=OKT(H):GOTO3010
```

A hang kitartására, ill. elengedésére biztosított, - a KI ill. SZ paraméterekkel meghatározott - idő programozása következik. Az előbbi értékét 0-20, az utóbbiét 0-10 között változtathatjuk.

```
1139 REM*** KI & SZ ******
1140 IFM-55THEN 1150
1141 KI(H)=KI(H)+1-21*INT(KI(H)/20)
1142 IFKI(H)THEN 1144
1143 POKE211,2+10*H:POKE214,14:SYS58640:PRINT" "
1144 A=10*H:B=14:C=KI(H):GOTO3010
1150 IFM-44THEN 1160
1151 SZ(H)=SZ(H)+1-11*INT(SZ(H)/10)
1152 IFSZ(H)THEN 1154
1153 POKE211,2+10*H:POKE214,15:SYS58640:PRINT" "
1154 A=10*H:B=15:C=SZ(H):GOTO3010
```

Ezután a szűrés kezelését oldjuk meg.

```
1159 REM*** MILYEN SZURES ***********
1160 IFM-22THEN 1162
1161 I=4:GOTO1170
1162 IFM-25THEN 1164
1163 I=5:GOTO1170
1164 IFM-30THEN 1166
1165 I=6:GOTO1170
1166 IFM-33THEN 1180
1167 I=7:GOTO1170
1170 IFME AND2↑ITHEN 1175
1171 ME=ME OR2↑I
1172 POKES+24,ME
1173 A=8:B=16+I:C$="■ ":GOTO3000
1175 ME=ME-2↑I
1176 POKES+24,ME
1177 A=8:B=16+I:C$="  ":GOTO3000
1179 REM*** MELYIK HANGOT *********
1180 IFM-45THEN 1182
1181 I=0:GOTO1190
1182 IFM-50THEN 1184
1183 I=1:GOTO1190
1184 IFM-53THEN 1186
1185 I=2:GOTO1190
1186 IFM-1THEN 1200
1187 I=3
1190 IFMH AND2↑ITHEN 1195
1191 MH=MH+2↑I
1192 POKES+23,MH
1193 A=2*I:B=19:C$="‾":GOTO3000
1195 MH=MH-2↑I
1196 POKES+23,MH
1197 A=2*I:B=19:C$=" ":GOTO3000
1199 REM*** SZURES FREKVENCIA *******
1200 IFM-14THEN 1205
1201 F1=F1+1-8*INT(F1/7)
1202 POKES+21,F1
1203 A=16:B=17:C=F1:GOTO3010
1205 IFM-17THEN 1215
1206 F2=F2+1-256*INT(F2/255)
1207 POKES+22,F2
1208 IFF2THEN 1210
1209 POKE211,16:POKE214,18:SYS58640:PRINT"||||"
1210 A=16:B=18:C=F2:GOTO3010
```

Az előbbi programrészben a SID 23. regiszterének tartalmát az MH változóban, 24. regiszterének tartalmát pedig az ME változóban kezeltük. Erről nem szabad majd megfeledkeznünk a hangerő, ill. a rezonancia beállításakor. Előbb azonban nézzük a szinkronizáció, körmoduláció, és a hang teljes leválasztásának kezelését!

```
1214 REM*** SZINK  KORMOD  X ********
1215 IFM-28THEN 1217
1216 I=1:GOTO1221
1217 IFM-39THEN 1219
1218 I=2:GOTO1221
1219 IFM-36THEN 1230
1220 I=3
1221 IFHF(H)AND2↑ITHEN 1225
1222 HF(H)=HF(H)+2↑I
1223 A=10*H+3+I:B=10:C$="‾":GOTO3000
1225 HF(H)=HF(H)-2↑I
1226 A=10*H+3+I:B=10:C$=" ":GOTO3000
```

Most már csak a rezonancia és a hangerő programozását kell megoldanunk.

```
1229 REM*** REZONANCIA ****************
1230 IFM-38THEN 1240
1231 SI=SI+1-16*INT(SI/15)
1232 IFSI THEN 1236
1233 MH=MH-240:POKES+23,MH
1234 POKE211,18:POKE214,20:SYS58640:PRINT"  "
1235 A=17:B=20:C=0:GOTO3010
1236 MH=MH+16:POKES+23,MH
1237 A=17:B=20:C=SI:GOTO3010
1239 REM*** HANGERO ***************
1240 IFM-40THEN 1250
1241 HE=HE+1-16*INT(HE/15)
1242 IFHE THEN 1246
1243 ME=ME-15 :POKES+24,ME
1244 POKE211,6 :POKE214,1 :SYS58640:PRINT"  "
1245 A= 5:B=1 :C=0:GOTO3010
1246 ME=ME+1 :POKES+24,ME
1247 A= 5:B=1 :C=HE:GOTO3010
1250 GOTO150
```

Feladat 16.

Ha az Olvasó még nem végezte volna el az utóbbi programrészletek alaposabb áttekintését, akkor azt ne mulassza el, legfőképp a rezonancia esetében.

Végre eljutottunk odáig, hogy a SID nyújtotta lehetőségeket kezelni tudjuk a programból. Bizonyos paraméterek beállítása,- mint pl. a hangerősségé, a hullámformáé, vagy a burkológörbéé, - mindenképpen szükséges ahhoz, hogy szintetizátorunk hallható legyen. Ezek kezdeti értéke ugyanis nulla. Egyedül az oktávot állítottuk 5-re a program elején (30-as sor). Ha valaki más paramétereknek is nullától különböző kezdeti értéket akar adni, ezt nyugodtan megteheti a 26-29 sorokban.

Térjünk rá a "zongorázás" programozására. Az természetesen nagyon hosszadalmas és lassú lenne, ha az egyes billentyűk állapotát külön-külön kérdeznénk le. Ehelyett a következőket tesszük: a HANG( ) tömböt úgy töltjük fel, hogy annak M. eleme azon billentyű hangsorbeli sorszámával egyezzen meg,

amelynek megnyomásakor a 197. byte értéke éppen M lesz. Egy példával gyorsan megvilágítjuk az előbbi kacifántos mondat jelentését.

```
3100 REM ***  HANG(I) ***
3101 DATA23,0,0,0,0,0,0,0,3,2,0,0,4,5,0
3102 DATA0,0,6,0,0,7,8,0,0,10,9,0,0,11
3103 DATA12,0,0,0,13,0,0,14,15,0,0,17
3104 DATA16,0,0,18,0,0,0,20,19,0,0,21
3105 DATA22,0,0,0,0,0,0,0,1,0,0
3106 REM ***  NEV$(I) ***
3107 DATA"B","H","C","CISZ","D","DISZ"
3108 DATA"E","F","FISZ","G","GISZ","A"
```

"Zongorázáskor" a következő billentyűket használhatjuk: Q,A,W,S,E,...,RETURN. Így a hangsorunkban a fenti billentyűk rendre az 1.,2.,3.,...23. helyet foglalják el. Ezért pl. HANG(62) = 1 mert a Q megnyomásakor a 197. byte értéke 62. Hasonlóan pl. HANG (10) = 2 vagy HANG(1) = 23. Abban az esetben, ha egy billentyű lenyomásakor nem szólal meg hang, a HANG-tömb megfelelő eleme 0 lesz.

A NEV$ (I) az I. hang nevét tartalmazza. Az első hang (Q) a H, a második (A) a C és így tovább.

M Feladat 17.

Miért van az, hogy az első hang a H, mégis a B az első adat a megfelelő DATA sorban?

Ezek után lássuk a megszólalást biztosító programrészt!

```
150 IFPEEK(653)THEN POKE646,8:GOTO 1000
151 POKE646,5
152 IFPEEK(197)=64THEN 152
153 M=PEEK(197)
154 IFM=48THEN 195
155 IFM-60THEN 159
156 P=1-P
157 POKE211,36:POKE214,17:SYS58640:PRINTP:FORI=1TO15:NEXTI
159 IFHANG(M)=0THEN 150
163 K=7*E:I=HANG(M)+12*OKT(H)
164 POKE211,4:POKE214,3:C$=NEV$(HANG(M))+"          "
165 SYS58640:PRINTC$
166 POKES-6+K,HB(I,0):POKES-7+K,HB(I,1)
167 POKES-3+K,HF(H)+1
168 IFP=1THENE=E+1-3*INT(E/3)
169 IFP=0THENE=H
170 FORI=1TO50*KI(H):NEXT
171 POKES-3+K,HF(H):FORI=1TOSZ(H)*25:NEXT
172 GOTO150
```

Számítógépünk két üzemmódban is dolgozhat, attól függően, hogy P=0 (MOD 0), vagy P=1 (MOD 1)*. A kettő közötti váltást a SPACE billentyű megnyomásával érhetjük el (155-157). A 157-es sorban egy üres ciklus található. Ennek hiányában a két üzemmód túl gyorsan váltakozna a SPACE billentyű lenyomásakor. P=1 esetén az E

* Ez utóbbi esetén a megszólaló hangok felváltva kerülnek az egyes oszcillátorokra.

változó értéke mutatja az éppen soron következő oszcillátor sorszámát, ami 0,1 vagy 2 lehet (168-169). Az E kezdeti értékét 1-nek kellett választanunk (30-as sor). Az I=HANG(M) + 12 * OKT(H) azt jelzi, hogy a megszólaltatni kívánt hangnak mi a HB tömb szerinti sorszáma (163-166). A hang kitartására és elengedésére biztosított időnél önkényesen választottam a szorzót 50-nek ill. 25-nek (170-171). Ezeket az értékeket tetszés szerint módosítani lehet. Figyeljük meg, hogy, ha P=1, akkor a H-nak megfelelő oszcillátor kontroll regiszteréhez tartozó értéket írjuk a másik két oszcillátor kontroll regiszterébe is. Így igen érdekes hatásokat érhetünk el, hiszen eközben mindegyik oszcillátor megtartja saját burkológörbéjét. A teljes program elkészülte után próbáljuk ki ezt az üzemmódot pl. úgy, hogy az egyes oszcillátorok burkológörbe paraméterei a következők legyenek: (0,1,13,5); (0,2,15,9); (0,0,11,14). Próbáljuk ki a körmodulációt és a szinkronizációt is!

Ha a £ jelet nyomjuk meg – azaz M=48 –, akkor a 154. sor hatására újabb üzemmódba jutunk.

```
190 REM **** AKKORD *****
195 POKE211,36:POKE214,17:SYS58640:PRINT"♥":FORI=1TO15:NEXTI
197 E=4:F=7:O=12*OKT(H)
200 M=PEEK(197):IFM=64THEN200
201 IFM=60THEN POKE646,8:GOTO 150
202 IFM-56THEN 205
203 E=4:F=7:K=0:GOTO200
205 IFM-59THEN 208
206 E=3:F=7:K=1:GOTO200
208 IFM-8 THEN 211
209 E=4:F=10:K=2:GOTO200
211 IFHANG(M)=0THEN 200
212 POKE646,5
213 ON K GOTO 216,219
214 POKE211,4:POKE214,3:C$=NEV$(HANG(M))+" DUR     "
215 SYS58640:PRINTC$:GOTO223
216 POKE211,4:POKE214,3:C$=NEV$(HANG(M))+" MOLL    "
217 SYS58640:PRINTC$:GOTO223
219 POKE211,4:POKE214,3:C$=NEV$(HANG(M))+"7        "
220 SYS58640:PRINTC$
223 I=HANG(M)+O:IFI>95-F THEN200
224 POKES-6+7*H,HB(I,0):POKES-7+7*H,HB(I,1)
226 POKES-6+7*H1,HB(I+E,0):POKES-7+7*H1,HB(I+E,1)
228 POKES-6+7*H2,HB(I+F,0):POKES-7+7*H2,HB(I+F,1)
230 POKES-3+7*H,HF(H)+1
232 POKES-3+7*H1,HF(H1)+1
234 POKES-3+7*H2,HF(H2)+1
236 FORI=1TO50*KI(H):NEXT
238 POKES-3+7*H,HF(H)
240 POKES-3+7*H1,HF(H1)
242 POKES-3+7*H2,HF(H2)
244 GOTO200
```

Itt háromféle akkord közül tudunk válogatni. Az 1 billentyű megnyomása után DÚR, a 2 billentyű megnyomása után MOLL, a 3-as megnyomása után DÚR7 akkordokat szólaltathatunk meg. Ebből az üzemmódból is a SPACE billentyű megnyomásával léphetünk ki. A dúr akkordok képzéséről volt már szó. Az alaphangot dó-nak tekintve a dúr akkodban a dó, mi, szó (0,4,7) hangok találhatók, a moll akkordban pedig az alaphangot lá-nak véve a lá, dó, mi, azaz (0,3,7) hangok. A DÚR7

akkordhoz valójában legalább négy hang kell, az egyszerű dúr akkordhoz tartozó dó, mi, szó és az ezt kiegészítő tá hang. Pl. az F7 akkordot az F, A, C és az ESZ=DISZ hangok alkotják. Mivel csak három oszcillátor áll rendelkezésünkre, kénytelenek vagyunk kicsit megcsonkítani az akkordot dó, mi, tá, azaz (0,4,10)-re. A programban K változó jelöli, hogy melyik akkord neve kerül kiírásra (203-220). Az Olvasó az eddigiek alapján kedve szerint bővítheti az akkord-választékot.

Természetesen csak akkor lesznek hallhatóak az akkordok, ha már mind a három oszcillátornál beállítottuk a megszólaláshoz szükséges paramétereket.

A programot kipróbálva bizonyára sokak igényeit nem elégíti ki az előbbi üzemmódok gyorsasága. A villámkezű billentyűsök kedvéért hoztam létre a következő üzemmódot, melyben minden fölösleges dologtól megszabadulva, néhány plusz ötlettel igyekeztem lehetővé tenni a gyors játékot.

```
1500 IFPEEK(653)-1 THEN 1546
1501 POKE211,37:POKE214,17:SYS58640:PRINT"2"
1502 O=12*OKT(H):K=7*H-7
1503 IFPEEK(197)=64THEN 1503
1504 M=PEEK(197)
1505 IFHANG(M)THEN 1508
1506 IFM-60THEN 1503
1507 GOTO1546
1508 I=HANG(M)+O
1509 POKES+K+1,HB(I,0):POKES+K,HB(I,1)
1510 POKES+K+4,HF(H)+1
1512 IFPEEK(197)=MTHEN 1512
1514 POKES+K+4,HF(H)
1516 GOTO1503
```

A legfontosabb eltérések az előzőektől:

1. A képerenyőre nem írunk ki semmit.
2. Nem használjuk a KI, ill. SZ paramétereket. Most minden hang pontosan addig szól, amíg a megfelelő billentyűt lenyomva tartjuk, a hang elengedésére semmi többlet időt nem biztosítunk (1509-1516).
3. Mivel az OKT(H) értéke nem változik a zenélési ciklusban, ezért a 12*OKT(H) értéket egy új változóban, az O-ban tároljuk, hogy ezt a szorzást ne kelljen minden hang megszólaltatásakor újra elvégezni. Hasonlóképpen a K = 7*H - 7 bevezetésével is időt nyerünk (1502).

Ezt az üzemmódot úgy tudjuk használni, hogy a SHIFT billentyűt lenyomva tartva játszunk a billentyűzeten. Kilépni úgy lehet, hogy a SHIFT-et felengedve lenyomjuk a SPACE billentyűt (1505-1507). Aki netán ezzel a gyorsasággal sem lenne megelégedve, annak ajánlom, hogy kísérletképpen minden program nélkül játsszon a billentyűkön, mint egy szintetizátoron, és ellenőrizze le, hogy minden leütött betű megjelent-e a képernyőn. Az eredmény bizonyára mindenkit meggyőz arról, hogy programunkat nem hibáztathatjuk a továbbiakban, ha néha kimarad egy-egy olyan hang, amit szerettünk volna megszólaltatni. Valószínűleg nem nyomtuk meg eléggé a megfelelő billentyűt.

```
1545 REM****  MEMORIA IRAS  **********
1546 IFPEEK(653)=6THEN 1600
1547 IFPEEK(197)=33THEN 1571
1548 POKE211,12:POKE214,23:SYS58640:PRINT"◣I▐"
1550 F=0:TI$="000000"
1551 IFPEEK(197)=64THEN 1551
1552 IFPEEK(197)=60THEN 1554
1553 AH(F)=60:GOTO1568
1554 ID(F)=TI:M=PEEK(197):AH(F)=M
1555 IFHANG(M)=0THEN 1551
1556 K=7*H:I=HANG(M)+12*OKT(H)
1558 POKES-6+K,HB(I,0):POKES-7+K,HB(I,1)
1560 POKES-3+K,HF(H)+1
1562 IFPEEK(197)=MTHEN 1562
1564 VE(F)=TI:POKES-3+K,HF(H):FORI=1TOSZ(H)*25:NEXT
1566 F=F+1:GOTO1551
1568 POKE646,8:POKE211,12:POKE214,23:SYS58640:PRINT"▟I▦"
1570 REM****  MEMORIA OLVASAS  ********
1571 IFPEEK(197)=38THEN 1546
1573 POKE211,15:POKE214,23:SYS58640:PRINT"◣O▐"
1575 F=0:TI$="000000"
1577 IFID(F)>TITHEN 1577
1578 M=AH(F):IFM=60THEN 1590
1579 IFHANG(M)=0THENF=F+1:GOTO1577
1580 K=7*H:I=HANG(M)+12*OKT(H)
1581 POKES-6+K,HB(I,0):POKES-7+K,HB(I,1)
1583 POKES-3+K,HF(H)+1
1585 IFVE(F)>TI THEN 1585
1587 POKES-3+K,HF(H):FORI=1TOSZ(H)*25:NEXT
1588 F=F+1:GOTO1577
1590 POKE646,8:POKE211,15:POKE214,23:SYS58640:PRINT"▟O▦"
```

A memória lehető legegyszerűbb változata a mostani. Egy dallam megjegyzésére képes, ezt részleteiben nem tudjuk átírni, mindig csak azt hallgathatjuk vissza, amit utoljára írtunk be. E változat áttanulmányozása után azonban bárki könnyedén átírhatja a programot, különleges kívánságokat is figyelembe véve.

Akkor tudunk egy dallamot a memóriába írni, ha a C=, a CTRL és az I billentyűt előzőleg egyszerre lenyomtuk (1546-1547). A dallam végét természetesen most is a SPACE billentyű megnyomásával jelezzük (1552-1553,1568). Ahhoz, hogy egy dallamot majd vissza tudjunk játszani, háromféle adatot kell tárolni:

- **a leütött billentyűket** (AH(F))
- **a billentyűk megnyomásának pillanatait** (ID(F))
- **a billentyűk elengedésének pilanatait** (VE(F)).

Az adattömböket a program elején 200-ra dimenzionáltuk (7-es sor), s így ha ezt nem növeljük, akkor dallamunk max. 200 hangból állhat. Az idő mérésével járó nehézségeket úgy tudjuk elkerülni, hogy az írás és az olvasás elején nullára állítjuk a TI$ **változót (1550,1575).**

Feladat 18.

a) Az Olvasó önállóan nézze át az egyes programsorok jelentését.

b) A memóriába íráskor a gyorsasággal lehetnek problémák. Írjuk át a fenti programrészletet a MOD 2 alapján úgy, hogy a lehető leggyorsabb legyen.

Már csak egy üzemmóddal kell megismerkednünk, az előre beprogramozott dalokkal, dallamokkal. Bár a képernyőn a program kilenc dallam megszólaltatásának lehetőségét jelzi, a könyv nyomdába adásáig ezek közül csak négy készült el. Billentyűzzük be az első kettőhöz tartozó DATA-sorokat!

```
3109 REM ***   R1   ****
3110 DATA 0,4,7,9,10,9,7,4
3111 DATA 0,4,7,9,10,9,7,4
3112 DATA 5,9,12,14,15,14,12,9
3113 DATA 0,4,7,9,10,9,7,4
3114 DATA 7,9,12,14
3115 DATA 5,7, 9,12
3116 DATA 7,9,12,14,31
3117 REM ***  P1  ****
3120 DATA 0, ,0, ,0, ,0,0
3121 DATA 0, ,0, ,0, ,0,0
3123 DATA 5, ,5, ,5, ,5,5
3124 DATA 0, ,0, ,0, ,0,0
3125 DATA -5, ,-5,-5
3126 DATA -2, ,-2,-2
3127 DATA  0, , 0, 0,7
3209 REM  ***  R2  P2  Q2  ***
3210 DATA 0,0,12,5,7,10, 7,5
3211 DATA 0,0,12,5,7,10,10,7
3212 DATA 0,0,12,5,7,10, 7,5
3213 DATA 0,0,12,5,7,10,10,7
3214 DATA 0,0,12,5,7,10, 7,5
3215 DATA 0,0,12,5,7,10,10,7
3216 DATA 0,0,12,5,7,10, 7,5
3217 DATA 0,0,12,5,7,10,10,7
3218 DATA 3,3, 3,2,2, 2, 1,1
3219 DATA 3,3, 3,2,2, 2, 1,1
3220 DATA 12,14,15,17,19,20,22,20
3221 DATA 19,20,19,19,19,19,19,19
3222 DATA 22,20,19,17,19,19,20,20
3223 DATA 19,19,19,19,19,19,19,19
3224 DATA 12,14,15,17,19,20,22,20
3225 DATA 19,19,19,19,19,19,19,19
3226 DATA 19,17,15,14,15,15,17,17
3227 DATA 19,19,19,19,19,19,19,19
3228 DATA 15,15,15,14,14,14,13,13
3229 DATA 15,15,15,14,14,14,13,13
3230 DATA 12,14,15,17,15,15,15,14
3231 DATA 15,15,15,15,15,15,15,15
3232 DATA 12,14,15,17,15,15,15,14
3233 DATA 12,12,12,12,12,12,12,12
3234 DATA 12,14,15,17,15,15,15,14
3235 DATA 15,15,15,15,15,15,15,15
3236 DATA 12,14,15,17,15,15,15,14
3237 DATA 12,12,12,12,12,12,12,12
3238 DATA  9,10,12, 9,10,12, 9,10
3239 DATA 12, 9,10,12,12,12,12,12
```

Ha a dallamokat meg akarjuk hallgatni, akkor a SHIFT és a CTRL billentyűt, valamint az 1,2,3, vagy 4 billentyűk valamelyikét kell egyszerre lenyomnunk. Ekkor egy ♠ karakter jelenik meg a képernyőn. Ezután egy olyan billentyűt kell leütnünk, amelyhez tartozik valamilyen hangmagasság (Q,A,W...), hogy meghatározzuk a dallam kezdőhangját. A zene persze csak akkor szólal meg, ha az oszcillátor paramétereket és a hangerőt előzetesen beállítottuk, ugyanis a dallamok a szintetizátor pillanatnyi, képernyőn látható paramétereinek megfelelő hangszínnel fognak elhangzani.

Az első dallamot játszó program érdekessége, hogy a zene eljátszása közben mi is zongorázhatunk a billentyűzeten. Ezt az 1616-1620. sorok biztosítják.

```
1600 IFPEEK(653)-5THEN 150
1601 POKE211,36:POKE214,17:SYS58640:PRINT4
1602 IFPEEK(197)=64THEN 1602
1603 M=PEEK(197):IFM-56THEN 1630
1604 POKE646,3:POKE211,21:POKE214,19:SYS58640:PRINT"♠":POKE646,8
1605 O=12*OKT(H):O2=12*OKT(H2):K=7*H:K1=7*H1:K2=7*H2
1606 IFPEEK(197)=60THENPOKE211,21:POKE214,19:SYS58640:PRINT" ":
     GOTO150
1607 IFPEEK(197)=64THEN 1606
1608 M=PEEK(197):IFHANG(M)=0THEN 1606
1609 I=HANG(M)+O
1610 FOR L=0TO44
1611 J=I+R1(L):J1=I+P1(L)
1612 POKES-6+K,HB(J,0):POKES-7+K,HB(J,1)
1613 POKES-6+K1,HB(J1,0):POKES-7+K1,HB(J1,1)
1614 POKES-3+K,HF(H)+1
1615 POKES-3+K1,HF(H1)+1
1616 TI$="000000"
1617 IFTI>10THEN 1621
1618 IFPEEK(197)=64THEN 1617
1619 J2=HANG(PEEK(197))+O2
1620 POKES-6+K2,HB(J2,0):POKES-7+K2,HB(J2,1):POKES-3+K2,HF(H2)+1:
     GOTO1617
1621 POKES-3+K,HF(H):POKES-3+K1,HF(H1):FORV=1TOSZ(H)*25
1622 NEXTV,L
1623 FORT=0TO24:NEXT T
1624 POKES-6+K,HB(J,0):POKES-7+K,HB(J,1)
1625 POKES-3+K,HF(H)+1
1626 FORT=1TO200*KI(H):NEXT T
1627 POKES-3+K,HF(H):FORV=1TOSZ(H)*25:NEXT V
1628 IFPEEK(197)=64THEN 1610
1629 GOTO1606
```

Ahhoz, hogy zongorajátékunk hallható legyen, a H2-höz tartozó oszcillátor kitartási szintjét (S) állítsuk magasabbra a másik két oszcillátor kitartási szintjénél.

A jelenlegi változatban — s ez a következő három beépített dallamra is igaz — a zene eljátszása után a program ellenőrzi, hogy van-e lenyomva billentyű. Ha nincs, akkor újrakezdi a dallamot (1628-as sor), ha van, akkor pedig a billentyű vizsgálata következik (1629). Úgy tudunk ebből az üzemmódból kilépni, ha a dallam befejezésének pillanatában nyomva tartjuk a SPACE billentyűt (1606). Amennyiben

olyan billentyűt nyomunk meg, amelyhez tartozik hang, akkor a dallam ismétlése ezen a hangon kezdődik (1608-1609).

Feladat 19.

Előfordulhat, hogy a billentyűzeten zongorázva nem vesszük észre a dallam végét, s így akkor is új hangról indul, amikor nem kellene. Módosítsuk a programot úgy, hogy a dallam lejátszása mindig ugyanazon a hangmagasságon kezdődjék!

A második előre programozott dallam saját szerzeményem. Bizonyára az Olvasók közül is sokan szeretnének többszólamú dallamokat kitalálni. Ehhez is nagy segítséget nyújt a számítógép. Tegyük fel például, hogy már megvan az alaptéma, vagyis az egyik szólam. Ekkor nincs más teendőnk, mint beprogramozni, és ciklusban újrajátszatni a géppel. Mialatt a gép folyamatosan ismétli dallamunkat, zongorázva megalkothatjuk a második szólamot. Ha elkészült, akkor ezt is gépre visszük, az első szólam mellé. Ettől fogva a két szólamot együtt ismételtetjük addig, amíg ki nem találjuk a harmadik szólamot is. Ezután már csak az egyes hangok hangszínét, az oszcillátorok paramétereit kell jól megválasztanunk. A második beprogramozott dallam a következő paraméterek mellett szól a legszebben:

OKT	OSC1	3 OSC 2	OSC3
A	0	0	0
D	0	0	0
S	15	15	14
R	14	10	14
WAVE	⊥	⊥	[illegible]+▼
PW L			0
PW H			1

```
1630 IFM-59THEN 1660
1631 POKE646,3:POKE211,23:POKE214,19:SYS58640:PRINT"♠":POKE646,8
1632 IFPEEK(197)=60THENPOKE211,23:POKE214,19:SYS58640:PRINT" ":
     GOTO150
```

```
1633 IFPEEK(197)=64THEN 1633
1634 M=PEEK(197)
1635 IFHANG(M)=0THEN 1633
1636 K=7*H:I=HANG(M)+12*OKT(H):K1=7*H1
1637 K2=7*H2
1638 FORL=0TO79
1639 J=I+R2(L):J1=I+P2(L):J2=I+Q2(L)
1640 POKES-6+K,HB(J,0):POKES-7+K,HB(J,1)
1641 POKES-6+K1,HB(J1,0):POKES-7+K1,HB(J1,1)
1642 POKES-6+K2,HB(J2,0):POKES-7+K2,HB(J2,1)
1643 POKES-3+K,HF(H)+1:POKES-3+K1,HF(H1)+1
1644 POKES-3+K2,HF(H2)+1
1645 FORT=1TO25*KI(H):NEXT T
1646 POKES-3+K,HF(H):POKES-3+K1,HF(H1):POKES-3+K2,HF(H2)
1647 FORV=1TOSZ(H)*25:NEXT V
1648 NEXT L
1649 FORT=0TO50:NEXT T
1650 IFPEEK(197)=64THEN 1638
1651 GOTO1632
```

Amennyiben több dallamot akarunk a szintetizátor memóriájában elhelyezni, akkor érdemes ezek számára egy szubrutint létrehozni, hogy a dallamjátszás megegyező részeit ne kelljen többször beírnunk.

```
1660 REM ******* 3
1661 IFM-8 THEN 1690
1662 POKE646,3:POKE211,25:POKE214,19:SYS58640:PRINT"↑":POKE646,8
1663 A=25:B=19:GOSUB4000
1664 FORL=0TO95
1665 J=I+P3(L):J1=I+P3(L):J2=I+P3(L)
1666 IFJ>95THENJ=0:J1=0:J2=0
1667 GOSUB 4020
1668 NEXT L
1669 IFPEEK(197)=64THEN 1664
1670 GOTO 1663
1690 REM ******* 4
1691 IFM-11THEN 150
1692 POKE646,3:POKE211,27:POKE214,19:SYS58640:PRINT"↑":POKE646,8
1693 A=27:B=19:GOSUB4000
1694 FORL=0TO95
1695 J=I+P4(L):J1=I+P4(L):J2=I+P4(L)+7
1696 IFJ2>95 THEN V1=0:V2=0:V3=0
1697 GOSUB 4020:V1=1:V2=1:V3=1
1698 NEXT L
1699 IFPEEK(197)=64THEN 1694
1700 GOTO 1693
1702 REM ******

4000 IFPEEK(197)=60THENPOKE211,A:POKE214,B:SYS58640:PRINT" ":GOTO150
4004 IFPEEK(197)=64THEN 4000
4006 M=PEEK(197)
4008 IFHANG(M)=0THEN 4004
4010 K=7*H:I=HANG(M)+12*OKT(H):K1=7*H1
4012 K2=7*H2
4014 RETURN
```

```
4020 IFV1THEN POKES-6+K,HB(J,0):POKES-7+K,HB(J,1)
4022 IFV2THEN POKES-6+K1,HB(J1,0):POKES-7+K1,HB(J1,1)
4024 IFV3THEN POKES-6+K2,HB(J2,0):POKES-7+K2,HB(J2,1)
4026 IFV1THEN POKES-3+K,HF(H)+1
4027 IFV2THEN POKES-3+K1,HF(H1)+1
4028 IFV3THEN POKES-3+K2,HF(H2)+1
4030 FORT=1TO25*KI(H):NEXT T
4032 POKES-3+K,HF(H):POKES-3+K1,HF(H1):POKES-3+K2,HF(H2)
4034 FORV=1TOSZ(H)*25:NEXT V
4036 RETURN
```

A harmadik dallam lejátszása előtt érdemes a paramétereknek a következő értékeket adnunk:

	OSC 1	OSC 2	2 OSC 3 •
A	0	0	0
D	0	0	0
S	14	14	15
R	10	7	7
WAVE	⊥	⊥	⊥

A negyedik dallam esetében pedig:

	OSC 1	2 OSC 2 •	OSC 3
A	0	0	0
D	1	2	2
S	13	11	11
R	6	13	6
WAVE	⊥	⊥	⊥

vagy:

	OSC 1	3 OSC 2 °	OSC 3
A	0	0	0
D	1	2	2
S	10	10	10
R	10	11	10
WAVE	⊥	⊥	⊥
KI		3	

```
3240 REM ******* P3
3241 DATA 5,5,5,3,0,0,3,4
3242 DATA 5,5,5,3,0,0,3,4
3243 DATA 5,5,5,3,0,0,3,4
3244 DATA 5,5,5,3,0,0,3,4
3245 DATA 10,10,10,8,5,5,8,9
3246 DATA 10,10,10,8,5,5,8,9
3247 DATA 5,5,5,3,0,0,3,4
3248 DATA 5.5,5,3,0,0,3,4
3249 DATA 12,12,12,12,12,12,12,11
3250 DATA 10,10,10,10,10,10,10,8
3251 DATA 5,5,5,5,5,5,5,5
3252 DATA 96,0,0,0,0,0,3,4
3260 REM ****BLUES/4/
3261 DATA 0,3,99,3,0,99,0,0,99,0,0,99
3262 DATA 0,3,99,3,0,99,0,0,99,0,0,99
3263 DATA 5,5,99,3,5,99,5,5,99,5,5,99
3264 DATA 0,3,99,3,0,99,0,0,99,0,0,99
3265 DATA 7,7,99,5,7,99,7,7,99,7,7,99
3266 DATA 7,7,99,6,5,99,5,5,99,5,5,99
3267 DATA 0,3,99,3,0,99,0,4,99,4,5,99
3268 DATA 5,6,99,6,7,7,7,7,7,7,7,99
```

Feladat 20.

Hallgassuk meg a paraméterértékek módosítása után is a beépített dallamokat!

MFeladat 21.

Mi a feladata az 1666-os és 1696-os sornak?

Feladat 22.

A 3. és 4. dallam alapján írjunk be újabb dallamokat a programba!

Zene és grafika

Bizonyára sok olvasóban felmerült már a kérdés, hogy lehet-e a zenélést grafikával kísérni, és ha igen, hogyan. Erre mutat be egy példát a fejezet. Hosszabb programról van szó, ezért a begépelése a lehetséges egyszerűsítések ellenére is sok időt igényel. Azt nem ígérhetem, hogy ekkora munka után mindenki el lesz ragadtatva az eredménytől, de annyi biztos, hogy a program áttanulmányozásából rengeteget lehet tanulni.

A zenét teljes egészében BASIC-ben írtam, s ezt egészítettem ki BASIC és SIMON'S BASIC nyelvű grafikai utasításokkal. Mivel ez a segédprogram széles körben elterjedt Magyarországon, remélem, senkinek sem okoz majd gondot a megszerzése.

(Ez a bekezdés CSAK MAGNÓSOKNAK szól. A SIMON'S BASIC és az egyszerű TURBO nem használható együtt. Az ABC TURBO működik mind a SIMON'S BASIC-kel, mind az egyszerű TURBO-val, viszont a SIMON'S BASIC-kel együtt használva nem tudjuk kimenteni programunkat. Ezért a következőt érdemes tenni: a SIMON'S BASIC-ben megírt programot először kimentjük, ezután betöltjük a TURBO-t. Az előbb kimentett programunkat beolvassuk, majd a TURBO-val mentjük ki (←S). Ezután jön az utolsó lépés: az ABC TURBO-t betöltjük a sima TURBO mellé, az utóbbival beolvassuk az előbb kimentett programot, majd ABC TURBO-val mentjük ki újra. Természetesen a SIMON'S BASIC-et is érdemes ABC TURBO-val kimenteni. Az elmondottak igen kacifántosnak tűnhetnek, s a "lemezesek" biztosan csak kacagnak az egészen, de a magnósoknak mindenképpen megéri ezt a módszert használni.

Azok, akik még nem ismerik a SIMON'S BASIC program nyújtotta lehetőségeket, az [1]-ben találhatnak részletes leírást. (Az általunk felhasznált grafikus utasításkészletről a 7.8 alfejezetben olvashatnak.) Programunkban szerepelni fognak sprite-ok is. Kezelésüket BASIC-ben oldjuk meg, annak ellenére, hogy erre is vannak SIMON'S BASIC utasítások. Ennek célja, hogy az Olvassó jobban megismerhesse a sprite-ok működését. (Ha valaki még nem programozott volna sprite-okat, akkor az ezzel kapcsolatos legfontosabb tudnivalókról az [1] 7.6-os alfejezetében olvashat.)

Most pedig röviden a program lényegéről. A grafika kezelésére két időszakasz áll rendelkezésre. Az egyik a Kapu-bit zárása és nyitása közötti idő-intervallum, a másik a kapu nyitása után a hang elengedésére biztosított időtartam. Arra kell csak ügyelnünk, hogy a dallamban ne legyenek zökkenők, ritmuskiesések. Ennek érdekében ismét a III. fejezet 5. példájában ismertetett szubrutint fogjuk használni, egy-egy dallamon belül pedig hasonló grafikai utasításokat alkalmazunk.

Egy ilyen program megírásakor a zenét és a grafikát nem kell feltétlenül egyszerre elkészítenünk. Jelen esetben pl. először a zene alakult ki fokozatosan, s csak azután találtam ki hozzá a grafikát.

Itt az ideje, hogy megismerkedjünk a programmal:

```
1 PRINT"[illegible]":POKE53280,3:POKE53281,3
2 V=53248:POKEV+21,0
3 POKE214,5:POKE211,13:SYS58640:PRINT"[illegible]COMMODORE MUSIC"
4 PRINT"[illegible]KESZITETTE: CSIKOS ZSOLT 1986"
6 PRINT"[illegible]JO SZORAKOZAST !!"
```

```
18 S=54272:O=1
19 FORI=0TO24:POKES+I,0:NEXT
20 DIM HB(95,1),A(160),B(160),C(160)
22 DIM B0(48),B1(48),C1(36),D1(128),D4(128),D2(8),D3(8),D7(8)
24 DIM D5(64),D6(32),E1(128),E2(128),E3(128)
30 FORI=1TO95
32 FR=262.8372*2↑(I/12)
34 HB(I,1)=INT(FR/256)
35 HB(I,0)=INT(FR-256*HB(I,1))
40 NEXT
50 FORI=1TO160:READX:A(I)=X+41:NEXT
53 FORI=1TO160:READX:B(I)=X+66:NEXT
56 FORI=1TO160:READX:C(I)=X+66:NEXT
59 FORI=1TO48:READX:B1(I)=X+41:NEXT
64 FORI=1TO36:READX:C1(I)=X+40:NEXT
66 FORI=1TO128:READX:D1(I)=X+64:NEXT
67 FORI=1TO8:READX:D2(I)=X+41:NEXT
68 FORI=1TO8:READX:D3(I)=X+41:NEXT
69 FORI=1TO8:READX:D7(I)=X+64:NEXT
71 FORI=1TO80:READX:D4(I)=X+40:NEXT
72 FORI=1TO32:READX:D5(I)=X+64:NEXT
73 FORI=1TO32:READX:D6(I)=X+79:NEXT
74 FORI=1TO128:READX:E1(I)=X+40:NEXT
75 FORI=1TO128:READX:E2(I)=X+47:NEXT
76 FORI=1TO128:READX:E3(I)=X+47:NEXT:X=0
77 FORI=0TO62:READY:POKE832+I,Y:NEXTI
78 FORI=0TO62:READY:POKE896+I,Y:NEXTI
79 FORI=0TO62:READY:POKE960+I,Y:NEXTI
```

Az 1-es sorban töröljük a képernyőt és beállítjuk a keret- és háttérszínt. A 2-es sorban a V változó a sprite-okat kezelő memóriarész kezdőcímével lesz egyenlő, majd a sprite-ok kikapcsolása, képernyőről való törlésük történik. A 18-as sorban S a SID kezdőcímével lesz egyenlő. A 19-es sorban töröljük a SID regisztereit. A 30-40 sorokban a szokásos módon feltöltjük a HB tömböt. Az 50-76 sorokban a zenéléshez szükséges tömböket töltjük fel, a 77-79 sorokban pedig a programban használt három sprite adatait helyezzük el a 13., 14., ill. 15. adatblokkban.

```
80 POKE2040,13:POKEV+39,8 :POKEV+28,7:POKEV+37,1:POKEV+38,0:
   POKEV+23,3
81 POKEV,85:POKEV+1,135
82 POKE2041,14:POKEV+40,8:POKEV+2,245:POKEV+3,131
83 POKE2042,15:POKEV+41,4:POKEV+4,150:POKEV+5,250
84 POKES+24,15
85 POKES+6,255
86 POKES+13,219
87 POKES+20,255:POKES+17,6
88 H1=32:H2=32:H3=16
89 GETA$:IFA$=""THENGOTO89
90 FORI=1TO128:K=A(I):N= 4:GOSUB1030:NEXT:POKES+13,207
91 FORI=1TO160:K=A(I):L=B(I):N=20:GOSUB1000:POKE1663+I,81:
   NEXT:POKES+13,175
92 FORI=1TO160:K=A(I):L=B(I):M=C(I):N=7:GOSUB1000:
   POKE1824-I,123:NEXT
93 POKES+13,191:PRINT"▣"
94 FORI=1TO160:K=A(I):L=B(I):M=C(I):N=7:GOSUB1000
95 POKE646,RND(1)*15:PRINT"▣ ";:NEXTI:PRINT"▩"
96 H3=64:POKES+20,175
```

```
97 FORI=1TO160:K=A(I):L=B(I):M=C(I):N=7:GOSUB1000:
   POKE1663+I,105:POKE1824-I,95
98 NEXTI
99 POKES+5,18:POKES+6,67:H1=32:POKEV+21,1
100 FORI=1TO160:K=A(I):L=B(I):M=C(I):N=7:GOSUB1000:
    POKEV,PEEK(V)+1:NEXTI
101 POKES+6,255
102 H1=32:H3=16:POKES+20,255
103 FORI=1TO160:K=A(I):L=B(I):M=C(I):N=7:GOSUB1000:
    POKEV+21,I:NEXTI
104 POKES+8,0:POKES+7,0
108 FORQ=1TO2:H2=32:H3=0:POKEV+21,1
109 FORI=1TO48:K=B1(I):L=A(I):M=0:N=7:GOSUB1000:
    POKEV+29,I:NEXTI,Q:M=0
110 H2=32:H3=0
111 FORI=1TO48:K=B1(I)+3:L=A(I)+3:M=0:N=7:GOSUB1000:
    POKEV+23,I:NEXTI:M=0
112 H2=32:H3=0:POKEV+1,155
113 FORI=1TO48:K=B1(I)+5:L=A(I)+5:M=0:N=7:GOSUB1000:
    POKEV+29,I:NEXTI:M=0
```

A 80-83 sorokban a sprite-okkal kapcsolatos byte-okat állítjuk be, míg a 84-88 sorokban a zenélés paraméterei kapnak értéket. A program az elindítás után kb. egy perc elteltével jut el idáig. Ha másoknak akarjuk megmutatni programunkat, akkor sincs baj, mert ha a bemutató előtt elindítjuk, akkor elvégzi az inicializálást, majd addig vár, amíg le nem nyomunk egy billentyűt (89. sor).

A programban két szubrutin található. Az elsőt (1000-1020) már ismerjük a korábbiakból, míg a második (1030-1060) ennek egy speciális kibővítése.

```
130 POKES+6,169:POKES+13,95:POKES+20,237:
    POKES+5,30:POKES+12,42
131 H1=128:H2=128:H3=128
132 FORQ=1TO5
133 K=RND(1)*50+20:L=0:M=0:N=RND(1)*3000:GOSUB1000:
    POKE53280,0:POKE53281,0
134 L=RND(1)*40+20:K=0:M=0:N=500:GOSUB1000:POKE53280,RND(1)*5:
    POKE53281,0
135 M=RND(1)*70 :K=0:L=0:N=900:GOSUB1000:POKE53280,0:POKE53281,0
136 FORJ=1TO5:LEFTW 7,17,10,10:NEXTJ,Q:FORJ=1TO5:LEFTW7,17,10,10:
    NEXTJ
1000 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1002 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1004 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1006 IFKTHENPOKES+4,H1+1
1008 IFLTHENPOKES+11,H2+1
1010 IFMTHENPOKES+18,H3+1
1012 FORT=1TON:NEXT
1014 IFKTHENPOKES+4,H1
1016 IFLTHENPOKES+11,H2
1018 IFMTHENPOKES+18,H3
1020 RETURN
1030 IFKTHENPOKES+1,HB(K,1):POKES,HB(K,0)
1032 IFLTHENPOKES+8,HB(L,1):POKES+7,HB(L,0)
1034 IFMTHENPOKES+15,HB(M,1):POKES+14,HB(M,0)
1036 IFKTHENPOKES+4,H1+1
1038 IFLTHENPOKES+11,H2+1
1040 IFMTHENPOKES+18,H3+1
1042 POKE214,RND(1)*5+14
1044 POKE211,RND(1)*6+17
1046 POKE646,RND(1)*16
1048 SYS58640:PRINT"■ "
1050 FORT=1TON:NEXT
1052 IFKTHENPOKES+4,H1
1054 IFLTHENPOKES+11,H2
1055 IFMTHENPOKES+18,H3
1060 RETURN
```

Az 1030-1060-as szubrutinban és a 95-ös sorban a véletlenszám-generátorral határozzuk meg a színeket, így ne csodálkozzunk, ha a program ismételt futtatásakor újabb színek jelennek meg a képernyőn. A 91-92-es sorokban nem új karaktereket nyomtatunk a képernyőre, csak a képernyőn lévő bizonyos karakterek alakját változtatjuk meg.

A 93-as sorban szereplő = HOME

A 95-ös sorban szereplő = CTRL+9 és = CTRL+0

A 99-es sorban bekapcsoljuk a nulladik sprite-ot.

A 100-as sorban történik ennek a sprite-nak a vízszintes irányú mozgatása.

Kis alakzatunk ugrálását, "táncát" a 103-as sorban programozzuk. Ennek lényege a következő: a figurához két sprite tartozik. Az első alapállásban, míg a második felemelt jobb lábbal. Az utóbbit egy kevéssel az előbbi fölé helyezzük, a program harmadik sprite-ját pedig a képernyőn kívüli, nem látható helyre. A 103-as sorban folytonosan változtatjuk, hogy melyik sprite legyen be- ill. kikapcsolva. Ez azt jelenti, hogy

amikor $I \equiv 1 \pmod 4$* akkor csak a 0. sprite látható,
amikor $I \equiv 2 \pmod 4$, akkor csak az 1. sprite,
amikor $I \equiv 3 \pmod 4$, akkor mindkettő,
amikor $I \equiv 0 \pmod 4$, akkor egyik sem.

A 108-as sor után csak a 0. sprite lesz bekapcsolva. A 109., 111., és a 113. sorokban ennek a sprite-nak az X, ill. Y tengely szerinti nagyítását kapcsoljuk ki illetve be, az I változó értékétől függően.

A 130-136. sorokban a fehér zaj és a véletlenszám-generátor segítségével érünk el érdekes hatást, melyet a program eddigi nézői közül néhányan sárkánynak, mások farkasnak véltek.

```
137 FORI=1TO1800:NEXT:POKEV+21,4:POKEV+38,3
138 POKE214,23:POKE211,10:SYS58640:PRINT"■ITT EGY VIRAG LESZ"
139 H1=80:H2=80:H3=80
140 J=1:Z=1:POKES+24,5
141 POKES+5,11:POKES+12,11:POKES+19,11
142 POKES+6,124:POKES+13,125:POKES+20,125
143 FORI=JTOJ+3STEP3:FLASH RND(1)*13+2,5
145 K=C1(I):L=0:M=0:N=100:GOSUB1000
146 L=C1(I+1):M=0:K=0:N=100:GOSUB1000
148 M=C1(I+2):L=0:K=0:N=80:GOSUB1000
149 POKEV+5,PEEK(V+5)-1:OFF:NEXTI
```

* $I \equiv 1 \pmod 4$ azt jelenti, hogy I néggyel osztva 1 maradékot eredményez.

```
150 Z=Z+1
151 IFZ<5THENJ=1:GOTO143
152 IFZ<7THEN J=7:GOTO143
153 IFZ<9THEN J=1:GOTO143
154 IFZ<11THEN J=7:GOTO143
155 IFZ<13THEN J=1:GOTO143
157 IFZ=13THEN J=13:GOTO143
158 IFZ=14THEN J=19:GOTO143
159 IFZ<17THEN J=1  :GOTO143
160 J=13
161 FORI=JTOJ+3STEP3
162 K=C1(I):L=0:M=0:N=200:GOSUB1000
163 L=C1(I+1):M=0:K=0:N=200:GOSUB1000
164 M=C1(I+2):L=0:K=0:N=170:GOSUB1000
165 NEXTI
166 Z=Z+1
169 IFZ=18THEN J=19:GOTO161
170 IFZ<21THEN J=1  :GOTO161
171 IFZ<23THEN J=25:GOTO161
172 IFZ<25THEN J=31:GOTO161
173 IFZ<27THEN J=25:GOTO161
174 IFZ<29THEN J=31:GOTO161
175 J=1:Z=1:F1=F1+1:POKE214,23:POKE211,24:SYS58640:PRINT"VOLT"
176 IFF1=1THENH1=16:H2=16:H3=16:POKEV+21,0:GOTO143
```

Ez a rész a program egyetlen "lassú" szakasza. Eredetileg gitáron találtam ki, s különböző akkordok felbontásából áll. Az akkordok így következnek egymás után: Am, Am, Am, Am, Am7, Am7, Am, Am, Am7, Am7, Am, Am, ||: D, F, Am, Am :|| ||: C^4, C^4, C, C :|| (ahol a C^4 akkord a C, F, G, C hangokból áll, szemben a C-dúr akkord C, E, G, C felbontásával).

Az egyes akkordokat a C1 tömb tárolja, és a Z, valamint a J változókkal szabályozzuk azt, hogy mikor melyik akkord szólaljon meg. E rész grafikai érdekessége az az azonos színű karakterek villogtatása a FLASH utasítás segítségével. A 149-es sor hatására fokozatosan jelenik meg a képernyőn a harmadik sprite, amit a 138-as sorban enyhe túlzással virágnak nevezünk.

```
180 HIRES5,1
198 POKES+5,0:POKES+6,255:H1=32
199 POKES+12,0:POKES+13,223:H2=32
200 POKES+19,0:H3=32
202 POKES+24,3:H1=16
203 FORI=1TO126:K=D1(I):L=0:M=0:N=90:GOSUB1000:
    LINE20,I,60,2*I,1:NEXTI
204 H1=16:H2=32:POKES+13,175:POKES+24,6
205 FORI=1TO126:K=D1(I):L=D2(F):M=0:N=60:GOSUB1000:F=F+1:
    IFF=9THENF=1
206 REC240-I,40+I,14,3*F,1:NEXTI:POKES+20,250:H3=128:NRM
207 FORI=1TO126:K=D1(I):L=D3(F):M=D7(F):N=30:GOSUB1000:
    F=F+1:IFF=9THENF=1
208 DOWNW0,0,40,25:NEXTI
209 POKES+6,250:POKES+13,250:H1=128:H2=128
210 FORQ=1TO3
211 FORI=1TO8:K=D7(I):L=D7(I):M=D7(I):N=120:GOSUB1000:NEXTI,Q
212 FORQ=1TO32:LEFTW 2,0,40,4
213 K=0:L=D3(F):M=0:N=130:GOSUB1000:F=F+1:IFF=9THENF=1
214 NEXTQ:H1=16:H2=32
215 FORI=1TO32 :K=D5(I):L=D3(F):M=0:N=120:GOSUB1000:F=F+1:
    IFF=9THENF=1
216 RIGHTW2,0,I+8,4:NEXTI:H3=32:POKES+20,249:FCHR 2,0,40,4,31
220 FORI=1TO32 :K=D5(I):L=D3(F):M=D6(I):N=90:GOSUB1000:F=F+1:
    IFF=9THENF=1
221 LEFTW RND(1)*4+2,0,40,1:NEXTI:X=X+1
222 UPW 2,0,40,4
223 IFX<3THENH3=20:POKES+24,9:GOTO220
224 IFX<7THENH2=20:GOTO220
225 H1=32:H2=32:H3=16
226 FORI=1TO32 :K=D5(I):L=D3(F):M=0:N=120:GOSUB1000:F=F+1:
    IFF=9THENF=1
227 NEXTI
229 POKE214,24:POKE211,10:SYS58640:PRINT"ES MEG NINCS VEGE "
230 FORI=1TO1000:NEXT:PRINT"□":HIRES0,1:MULTI5,6,7
231 POKES+13,189:POKES+20,189:POKES+24,12
232 H1=20:H3=32
234 FORQ=1TO2
235 FORI=1TO32 :K=E1(I):L=0:M=0:N=70:GOSUB1000:LINE 5↑Q,50,70,2*I,Q
236 NEXTI:NEXTQ:H1=32
237 FORI=1TO128:K=E1(I):L=E2(I):M=E3(I):N=65:GOSUB1000:
    LINE 50+I,16,90,50,2
238 NEXTI
239 FORI=1TO128:K=E1(I):L=E2(I):M=E3(I):N=65:GOSUB1000:
    LINE 40+I,50,100,10,4
240 NEXTI
241 TEXT50,65,"ZSOLT'▒S MAGIC",1,1,6
242 TEXT50,76,"▒SAYS GOOD-BYE",2,1,6
243 TEXT 64,87,"▒TO YOU ",1,1,8
244 TEXT 52,120,"BYE-BYE",4,1,11
245 GOTO245
280 END
```

Ebben a szakaszban kétszer is használjuk a bit-térképes üzemmódot. Először (180-206) egyenesek és téglalapok rajzolásával érünk el látványos eredményt. Különösen az egyenesek megjelenítésekor nem pontosan az kerül a képernyőre, aminek elméletileg kellene. Persze, ha belegondolunk, hogy ezek nem valódi egyenesek, hanem csak egy négyzetrács elemei, akkor már érthető lesz a dolog.

A következőkben kicsit elvonttá válik a zene, ami talán nem mindenkinek tetszik. A hatás a 223,224 sorokban a körmoduláció bekapcsolásával csak fokozódik.

A finálé a 230-as sorban kezdődik, amikor a többszínű bittérképes üzemmódot használjuk. Figyeljük meg a ‹plot-type› változását az egyes esetekben.

A 241-244-es sorok hatására a program egy szöveget ír a képernyőre.

Az itt szereplő [B] = CTRL+B,

és a [A] = CTRL+A

Most már csak az adatokat kell megismernünk:

```
300 DATA 3,3,10,3,3,10,11,10
301 DATA 3,3,10,3,3,10,11,10
302 DATA 1,1,8,1,1,8,9,8
303 DATA 1,1,8,1,1,8,9,8
304 DATA 3,3,10,3,3,10,11,10
305 DATA 3,3,10,3,3,10,11,10
306 DATA 1,1,8,1,1,8,9,8
307 DATA 1,1,8,1,1,8,9,8
308 DATA 3,3,10,3,3,10,11,10
309 DATA 3,3,10,3,3,10,11,10
310 DATA 1,1,8,1,1,8,9,8
311 DATA 1,1,8,1,1,8,9,8
312 DATA 3,3,10,3,3,10,11,10
313 DATA 3,3,10,3,3,10,11,10
314 DATA 1,1,8,1,1,8,9,8
315 DATA 1,1,8,1,1,8,9,8
316 DATA 3,5,6,8,6,5,3,1
318 DATA 3,5,6,8,6,5,3,1
320 DATA 8,6,3,5,6,3,1,0
322 DATA 1,3,3,3,3,-41,3,3
325 REM*************
340 DATA 5,5,5,5,5,5,4,4
342 DATA 2,2,2,2,2,2,0,2
344 DATA 4,4,4,4,4,4,2,2
346 DATA 0,0,0,0,0,0,0,0
348 DATA 5,5,5,5,5,5,4,4
350 DATA 2,2,2,2,2,2,0,2
352 DATA 4,4,4,4,4,4,2,2
354 DATA 0,0,0,0,0,0,0,0
356 DATA 4,4,4,4,4,4,5,5
358 DATA 7,7,7,7,7,7,5,4
360 DATA 5,5,5,5,5,5,4,4
362 DATA 0,4,4,4,4,4,4,4
364 DATA 5,5,5,5,5,5,7,7
366 DATA 9,9,9,9,9,9,10,9
368 DATA 7,7,7,7,7,7,4,4
370 DATA 0,0,0,0,0,0,0,0
372 DATA 0,2,5,2,2,0,2,2
374 DATA 0,2,5,2,2,0,2,2
376 DATA 0,2,5,7,7,5,7,7
378 DATA 5,5,2,2,2,2,2,2
380 REM*************
400 DATA 4,5,4,7,5,4,5,4
402 DATA 4,5,4,7,5,4,5,4
404 DATA 7,5,4,7,5,4,7,5
406 DATA 4,5,5,5,5,5,4,4
408 DATA 4,5,4,7,5,4,5,4
410 DATA 4,5,4,7,5,4,5,4
412 DATA 7,5,4,7,5,4,7,5
414 DATA 4,5,5,5,5,5,4,4
416 DATA 4,5,7,9,10,9,12,10
418 DATA 9,7,9,10,9,7,5,4
420 DATA 7,5,4,5,4,0,2,2
422 DATA 2,-66,2,2,2,0,0,0
424 DATA 4,5,7,9,10,9,12,10
426 DATA 9,7,9,10,9,7,5,4
428 DATA 7,5,4,5,4,0,2,2
430 DATA 2,-66,2,2,2,0,0,0
432 DATA -3,-3,-3,-3,-3,-3,-5,-5
434 DATA 5,5,5,5,5,5,4,4
436 DATA -7,-7,-7,-7,-7,-7,-8,-8
438 DATA 2,2,2,2,2,2,2,2
500 DATA 3,3,3,5,6,5
501 DATA 3,3,3,5,6,5
502 DATA 3,3,3,5,6,5
503 DATA 8,6,5,6,5,1
506 DATA 3,3,3,5,6,5
507 DATA 3,3,3,5,6,5
508 DATA 3,3,3,5,6,5
509 DATA 1,1,1,1,1,1
```

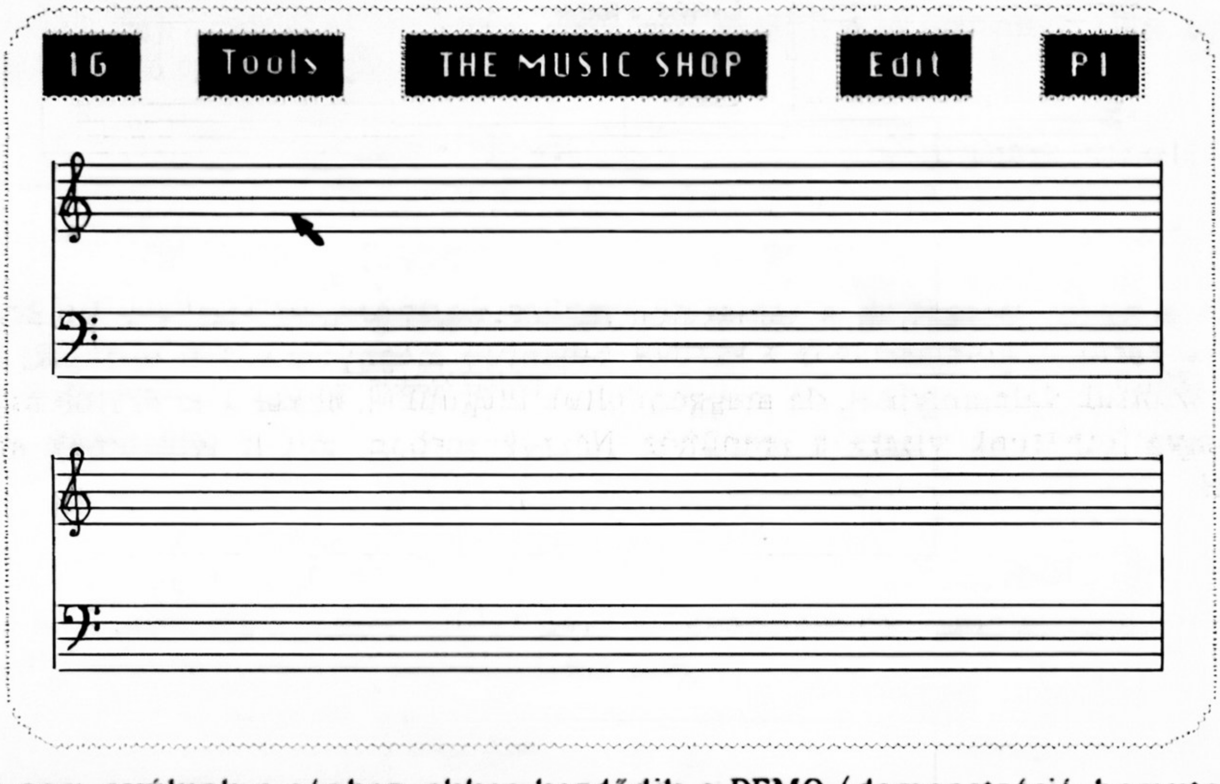
```
549 REM* * * * * * * * * * * * *
550 DATA 0,5,8,12,8,5
551 DATA 0,5,8,15,8,5
552 DATA-2,5,10,14,10,5
553 DATA 1,5,8,13,8,5
554 DATA-4,1,3,8,3,1
555 DATA-4,0,3,8,3,0
556 REM* * * * * * * * * * * * * *
570 DATA4,5,12,4,5,12,7,2
571 DATA4,5,12,4,5,12,7,2
572 DATA4,5,12,4,5,12,7,2
573 DATA4,5,12,4,5,12,7,2
574 DATA 0,1,2,7,6,2,0,2
575 DATA 0,1,2,7,6,2,0,0
576 DATA 0,1,2,7,6,2,0,2
577 DATA 0,1,2,7,6,2,0,0
578 DATA 14,8,3,8,12,8,6,8
579 DATA 12,13,12,8,6,8,12,8
580 DATA 14,8,3,8,12,8,6,8
581 DATA 12,13,12,8,6,8,12,8
582 DATA 12,6,1,6,10,6,4,6
583 DATA 10,11,10,6,4,6,10,6
584 DATA 12,6,1,6,10,6,4,6
585 DATA 10,11,10,6,4,6,10,6
586 REM* * * * * * * * * * *
590 DATA 0,2,5,2,2,5,7,7
592 DATA 0,1,5,1,0,1,7,1
595 DATA 0,0,0,0,-64,-64,-64,-64
596 REM**********************
597 DATA 0,0,0,0,0,0,0,0
598 DATA 0,0,0,0,0,0,0,0
599 DATA 0,0,0,0,0,0,0,0
600 DATA 0,0,0,0,0,0,0,0
601 DATA 1,1,1,1,1,1,1,1
602 DATA 1,1,1,1,1,1,1,1
603 DATA 1,1,1,1,1,1,1,1
604 DATA 1,1,1,1,1,1,1,1
605 DATA 1,3,1,3,1,3,1,3
606 DATA 1,3,1,3,1,3,1,3
620 REM******************
621 DATA 5,5,5,5,5,5,5,5
622 DATA 4,4,4,4,4,4,4,4
623 DATA 3,3,3,3,3,3,3,3
624 DATA 1,1,1,1,1,1,1,1
630 DATA 3,-79,-79,-79,3,-79,3,-79
631 DATA 2,2,2,2,2,2,2,2
632 DATA 1,-79,-79,-79,1,-79,1,-79
633 DATA 0,0,0,0,0,0,0,0
634 REM********
650 DATA 24,18,19,23,19,18,17,15
651 DATA 24,18,19,23,19,18,17,15
652 DATA 24,18,19,23,19,18,17,15
653 DATA 14,10,9,5,7,-40,7,-40
654 DATA 24,18,19,23,19,18,17,15
655 DATA 24,18,19,23,19,18,17,15
656 DATA 24,18,19,23,19,18,17,15
657 DATA 14,10,9,5,7,-40,7,-40
658 DATA 10,14,10,9,5,4,0,4
659 DATA 10,14,10,9,5,-40,5,-40
660 DATA 10,14,10,9,5,4,0,4
661 DATA 10,14,10,9,7,-40,7,-40
662 DATA 7,8,9,10,12,13,14,15
663 DATA 7,8,9,10,12,13,14,15
664 DATA 7,8,9,10,12,13,14,15
665 DATA 16,17,18,19,24,-40,24,-40
666 REM************
670 DATA 8,-47,-47,-47,-47,-47,7,-47
671 DATA 8,-47,-47,-47,-47,-47,7,-47
672 DATA 10,-47,-47,-47,-47,-47,8,-47
673 DATA  7,-47,-47,-47,-47,-47,-47,-47
674 DATA 8,-47,-47,-47,-47,-47,7,-47
675 DATA 8,-47,-47,-47,-47,-47,7,-47
676 DATA 10,-47,-47,-47,-47,-47,8,-47
677 DATA  7,-47,-47,-47,-47,-47,-47,-47
678 DATA  0,-47,-47,-47,-47,-47,-47,-47
679 DATA  0,-47,-47,-47,-47,-47,-47,-47
680 DATA  0,-47,-47,-47,-47,-47,-47,-47
681 DATA  0,-47,-47,-47,-47,-47,-47,-47
682 DATA 1,-47,-47,-47,-47,-47,0,-47
683 DATA 1,-47,-47,-47,-47,-47,3,-47
684 DATA  0,-47,-47,-47,-47,-47,-47,-47
685 DATA  0,-47,-47,-47,-47,-47,-47,-47
```

```
686 REM*******
690 DATA 12,-47,-47,-47,-47,-47,10,-47
691 DATA 12,-47,-47,-47,-47,-47,10,-47
692 DATA 13,-47,-47,-47,-47,-47,12,-47
693 DATA 10,-47,-47,-47,-47,-47,-47,-47
694 DATA 12,-47,-47,-47,-47,-47,10,-47
695 DATA 12,-47,-47,-47,-47,-47,10,-47
696 DATA 13,-47,-47,-47,-47,-47,12,-47
697 DATA 10,-47,-47,-47,-47,-47,-47,-47
698 DATA  7,-47,-47,-47,-47,-47,-47,-47
699 DATA  7,-47,-47,-47,-47,-47,-47,-47
700 DATA 7,-47,-47,-47,-47,-47,-47,-47
701 DATA  7,-47,-47,-47,-47,-47,-47,-47
702 DATA 8,-47,-47,-47,-47,-47,7,-47
703 DATA 8,-47,-47,-47,-47,-47,10,-47
704 DATA  7,-47,-47,-47,-47,-47,-47,-47
705 DATA  7,-47,-47,-47,-47,-47,-47,-47
800 REM ****** SPRITE ***
802 DATA  3,255,0,  15,255,192
803 DATA 13,85,192,    13,153,192
804 DATA 13,101,192,    15,88,192
805 DATA 170,170,168
806 DATA 170,170,168
807 DATA 170,170,168
808 DATA 162,170,40
809 DATA 162,170,40
810 DATA 162,170,40
811 DATA 162,170,40
812 DATA  83,255,20
813 DATA   3,255, 0
814 DATA  15,255,192
815 DATA  63,255,240
816 DATA  63,255,240
817 DATA 255,255,252
818 DATA   1,  1, 0
819 DATA  15,207,192
820 REM ****** SPRITE 2 ***
822 DATA  3,255,0,  15,255,192
823 DATA 13,85,192,    13,153,192
824 DATA 13,101,192,    15,88,192
825 DATA 170,170,168
826 DATA 170,170,168
827 DATA 170,170,168
828 DATA 162,170,40
829 DATA 162,170,40
830 DATA 162,170,40
831 DATA 162,170,40
832 DATA  83,255,20
833 DATA  15,255, 0
834 DATA   1,255,192
835 DATA   1, 63,240
836 DATA  15,207,240
837 DATA   0, 15,240
838 DATA   0,  1, 0
839 DATA   0, 15,192
850 REM *** SPRITE "VIRAG" *****
851 DATA 0,0,0
852 DATA 0,32,0
853 DATA 0,32,0
854 DATA 0,84,0
855 DATA 10,87,240
856 DATA 32,84,48
857 DATA  0,32,48
858 DATA  0,35,192
859 DATA  0,140,0
860 DATA 128,12,0
861 DATA 168,15,192
862 DATA  42,128,192
863 DATA 10,171,192
864 DATA  2,175,0
865 DATA  0,188,0
866 DATA  3,240,0
867 DATA 15,192,0
868 DATA 12,0,0
869 DATA 15,0,0
870 DATA  3,192,0
871 DATA  0,192,0
```

A "Music Shop" program

A fejezet egy olyan programot ismertet, mely igen elterjedt hazánkban, de bizonyára sokan vannak, akik nem tudják kezelni. Miért éppen ezt választottuk a sok zenei segédprogram közül? - kérdezheti az Olvasó. Nos, a döntő érv a Music Shop mellett az volt, hogy viszonylag egyszerűen kezelhető, szépen megtervezett program. Persze, a külső forma és a kezelhetőség mellett az sem elhanyagolható szempont, hogy mit tud a program. A Music Shop egy tetszőleges, általunk megírt kotta lejátszására készült. Emellett azonban számos egyéb dologra is képes. Pl. a kottát ki tudja rajzolni a nyomtatóra, el tudja menteni lemezre. Sok dalt már eredetileg beépítettek a programba. A zene lejátszása közben is élni tudunk a C-64 által biztosított lehetőségekkel (ADSR, szűrés, stb.), sőt, előre beprogramozott "hangszereket" is használhatunk.

A LOAD "MUSIC SHOP",8,1 utasítás segítségével tölthetjük be a programot, mely ezután elindul. Néhány másodperc után a PLEASE WAIT... (KÉREM VÁRJON) felirat jelenik meg a képernyőn. Ezután bemutatkozik a program, majd rövid idő múlva a következő kép kerül a képernyőre:

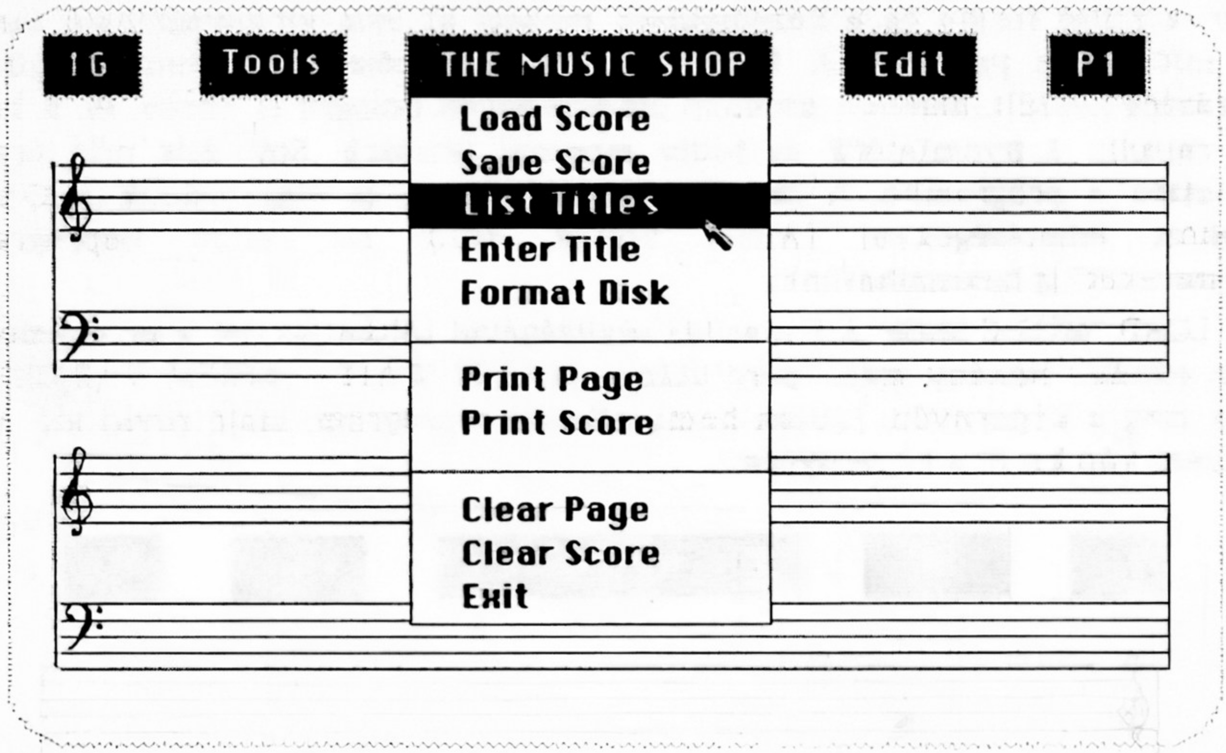


Ha nem nyúlunk a géphez, akkor kezdődik a DEMO (demonstráció=bemutató). Ez olyan, mintha egy lemezboltban meghallgatnánk az összes lemezt, hiszen egymást követik a programhoz tartozó jól ismert melódiák.

A DEMO-ból a RUN/STOP billentyű megnyomásával léphetünk ki. Ha valaki ismét el szeretné indítani a DEMO-t, ezt megteheti a C= és a D billentyűk együttes lenyomásával, nem kell újra betöltenie a programot.

Fontos a képernyőn látható fekete nyíl szerepe. Ezt ugyanúgy tudjuk mozgatni, mint programíráskor a kurzort. (Igaz, van más lehetőség is, pl.: C= +S = CTRL+A = SHIFT+RUN/STOP = NYÍL LE (ez utóbbiakat természetesen felesleges megjegyezni.)*

Mozgassuk a nyilat közvetlenül a THE MUSIC SHOP felirat alá, majd nyomjuk meg a RETURN billentyűt. Ekkor a főmenü táblázata jelenik meg a képernyőn:



Ha a nyilat mozgatjuk a táblázaton, akkor az éppen választható funkció címe keretbe kerül. A kiválasztás is a RETURN billentyű megnyomásával történik. Ha már kiválasztottuk valamelyiket, de meggondoltuk magunkat, akkor a RUN/STOP billentyűt lenyomva juthatunk vissza a menühöz. Nézzük sorban, mit is jelentenek az egyes címek!

* A program joystick-kal is működtethető, de mivel a billentyűzetről kényelmesebb kezelni, ezért csak ezt ismertetjük.

Load Score (Partitúra betöltés)

Ennek választásakor a következő táblázat jelenik meg:

DISK ACCESS	Diszk-művelet
Enter Title:	Add meg a címet!
Load **Cancel**	Betöltés Stornó

Ebben az esetben tehát egy tetszőleges, a lemezen megtalálható dalt tölthetünk a gépbe. A dal címét a billentyűzetről kell begépelnünk. Ha netán elütnénk a címet, akkor az INST/DEL billentyűvel korrigálni tudjuk. Begépelés után a nyilat vagy a Load, vagy a Cancel feliratú téglalapra kell mozgatni. Ha úgy választottunk, hogy töltse be az illető dalt, akkor ismét két dolog lehetséges:

1. Létezik egy ilyen nevű program, és a gép betölti. A töltés ideje alatt egy, az alábbihoz hasonló táblázat látható:

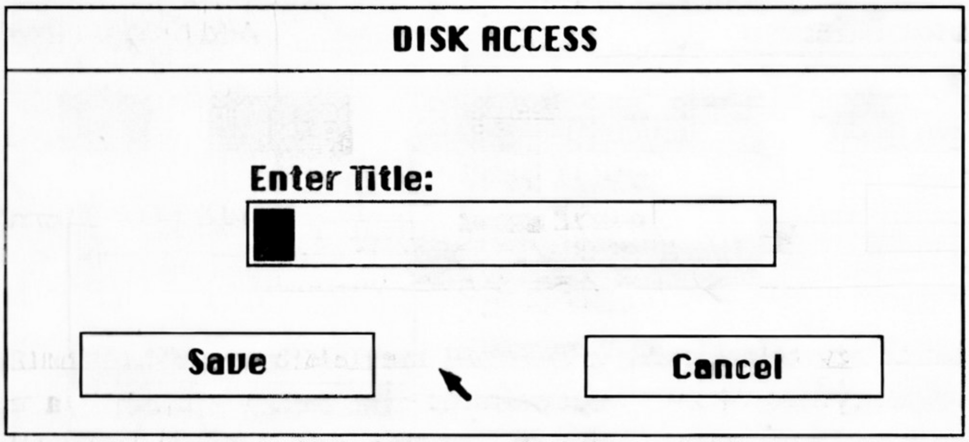
DISK ACCESS	Diszk-művelet
Loading GREENSLEEVES	A ... betöltése
Please Wait	Kis türelmet!

2. Ha nem található a lemezen ilyen nevű szekvenciális file, azaz nincs ilyen dal a boltunkban, akkor a következő felirat jelenik meg:

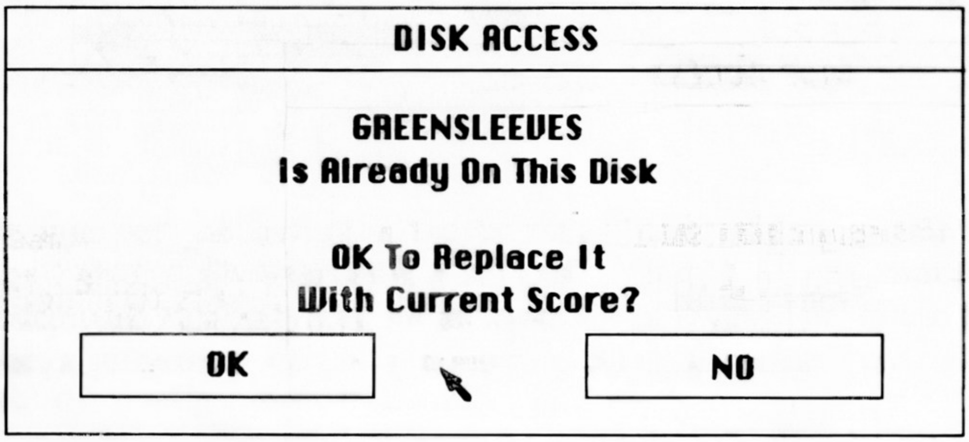
PROBLEM	Diszk-művelet
Can't Find GREENS	Nem találom a ...-t
OK	

Csak akkor tudunk visszatérni a programhoz, ha a problémát nyugtáztuk, azaz a nyilat az OK keretére mozgatjuk, és megnyomjuk a RETURN billentyűt, vagy a nyíl helyzetétől függetlenül a RUN/STOP billentyűt nyomjuk meg.

Save Score (Partitúra lemezre mentése)



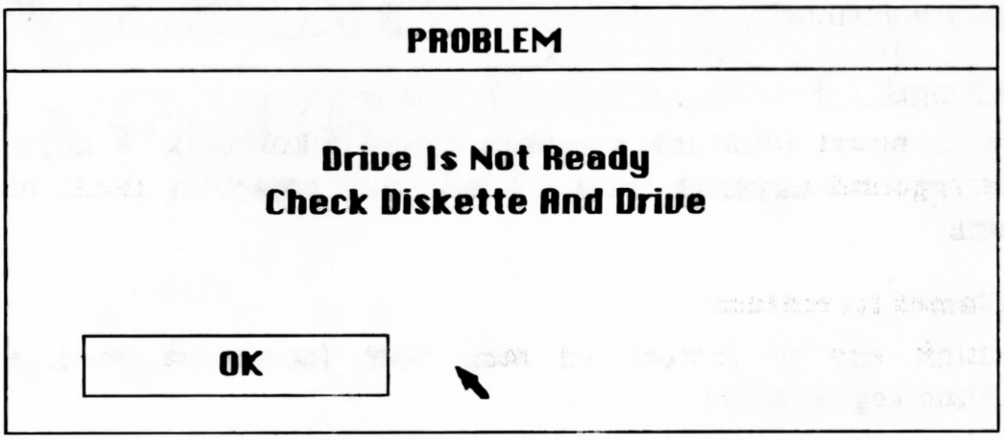
A gépben jelenleg megtalálható kottát, partitúrát tudjuk lemezre menteni tetszőleges (max. 16 karakter hosszú) címmel. Amennyiben már szerepel ilyen nevű dal a lemezen, akkor a következő táblázat jelenik meg:



A ...
már rajta van
ezen a lemezen
Átírhatom a
jelenlegivel?

Ha nem akarjuk a már meglevő programot a mostanival felülírni, akkor a NO feliratú téglalapra kell vinni a nyilat, s a RETURN-t megnyomni.

Amennyiben a kotta kimentése valamilyen akadályba ütközik, - pl. nincs lemez a meghajtóban -, akkor az alábbi szöveg olvasható a képernyőn:



Ellenőrizd a lemezt és a lemezegységet!

A probléma tudomásulvételét legegyszerűbben ismét a RUN/STOP billentyű lenyomásával jelezhetjük.

List Titles (Címek listázása)

Egy jó lemezboltban nyilvántartják a lemezkínálatot. Ez a mi boltunkban sincs másképp.



Természetesen nagy választék esetén nem férnek el egy lapon a címek. A fekete nyilat a fehér nyílhoz téve tudunk lapozni a címlistán, csak a RETURN billentyűt kell megnyomni. Az utolsó oldalt ismét az első követi, így nincs szükség visszalapozásra.

Ha kiválasztottuk a kívánt dal címét, akkor nincs más teendő, mint rámozgatni a fekete nyilat, és megnyomni a RETURN-t. Mivel most nem kell a címet begépelnünk, ezért a Load Score helyett érdemesebb ezt a változatot használni egy dallam betöltésekor.

E funkcióból a RUN/STOP billentyű megnyomásán kívül úgy is kiléphetünk, hogy a fekete nyilat a Score Titles felirat előtti kis négyzetre, vagy a táblázaton kívülre visszük, majd lenyomjuk a RETURN-t.

Enter Title (Adj címet)

Ennek segítségével új nevet adhatunk a gépben szereplő kottának. A képernyőn megjelenő táblázat lényegében ugyanaz, mint a Load Score esetében, tehát most is ugyanúgy kell eljárnunk.

Format Disk (Új lemez formázása)

Amennyiben vettünk egy új lemezt, és még nem formáztuk meg, ezt is megtehetjük a Music Shop segítségével.

DISK ACCESS

Insert Diskette
To Be Formatted
WARNING
Entire Disk Will Be Erased!

Format | **Cancel**

Tedd be a formattálni kívánt lemezt! FIGYELMEZTETÉS A lemez teljes tartalma törlődik!

Erre a szolgáltatásra akkor lehet szükség, ha szerzeményeinket nem arra a lemezre akarjuk kimenteni, amelyik a Music Shop programot tartalmazza – hiszen azon nem sok szabad hely van.

Print Page

Az éppen gépben lévő kotta képernyőn látható oldalának printerrel való kinyomtatása.

Print Score

A teljes kotta kinyomtatása. Probléma esetén a következő táblázat jelenik meg:

PROBLEM

Printer Is Not Ready
Check Printer And Connections

OK

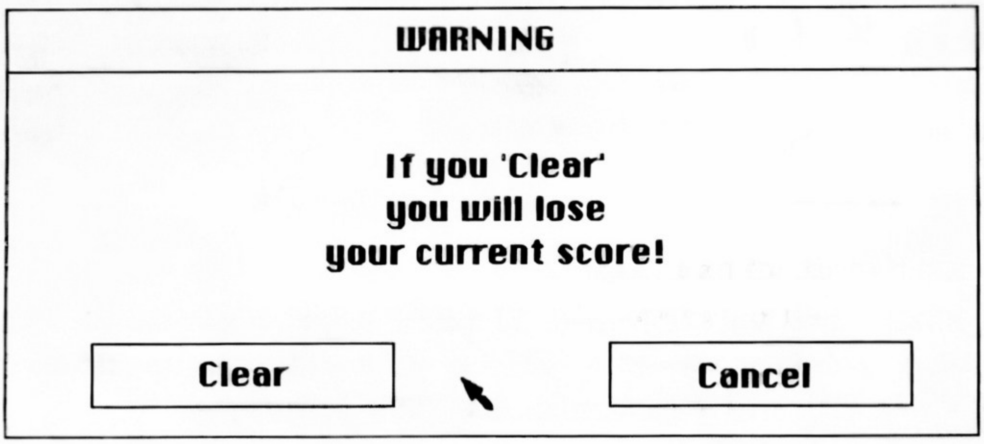
Ellenőrizd a nyomtatót és a csatlakozókat!

Clear Page

Az éppen gépben lévő kotta éppen látható oldalának törlése. A gyakorlatban e szolgáltatás használata helyett kényelmesebb a SHIFT és a CLR/HOME együttes lenyomásával törölni a képernyőn látható kottarészt.

Clear Score

A gépben lévő teljes partitúra törlése.

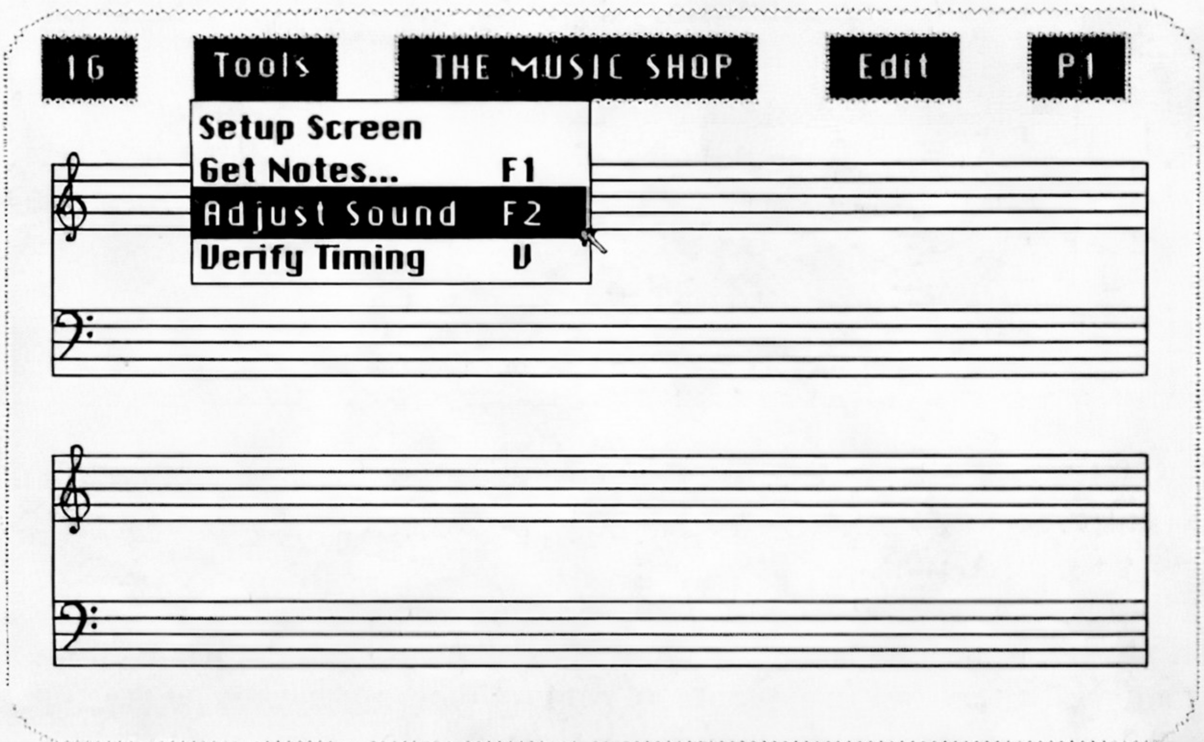


'Törlés' esetén az aktuális kotta elvész!

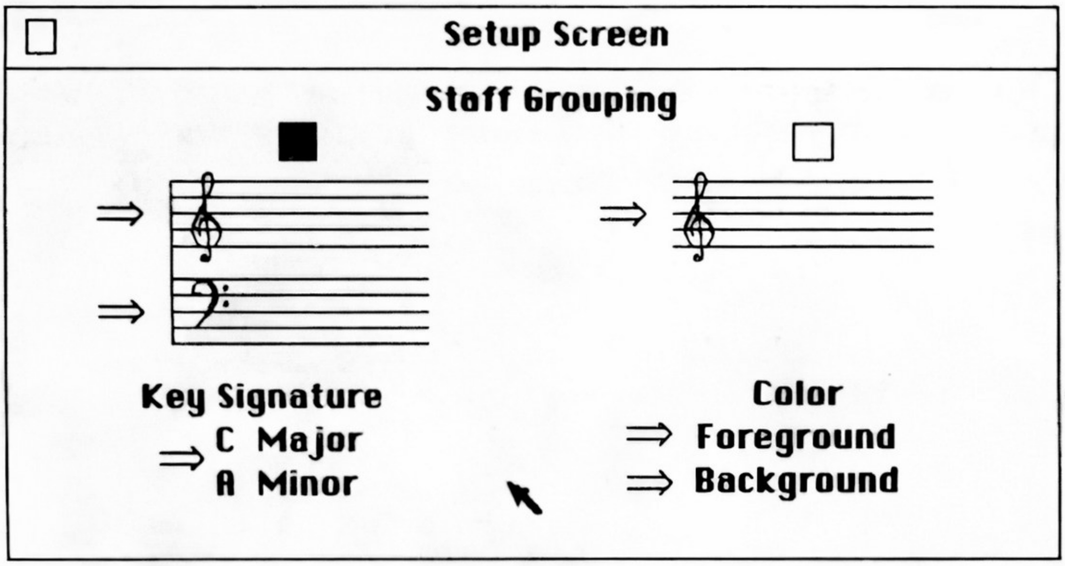
Exit

Kilépés a Music Shop-ból

A zenebolt szolgáltatásainak megismerése után vigyük a nyilat a Tools (a zenélés eszközei) felirat alá, s nyomjuk meg a RETURN-t! Ekkor a következő táblázat jelenik meg:



Ismét nézzük sorrendben az egyes szolgáltatásokat!



Képernyő-beállítás
Kottavonalak csoportosítása

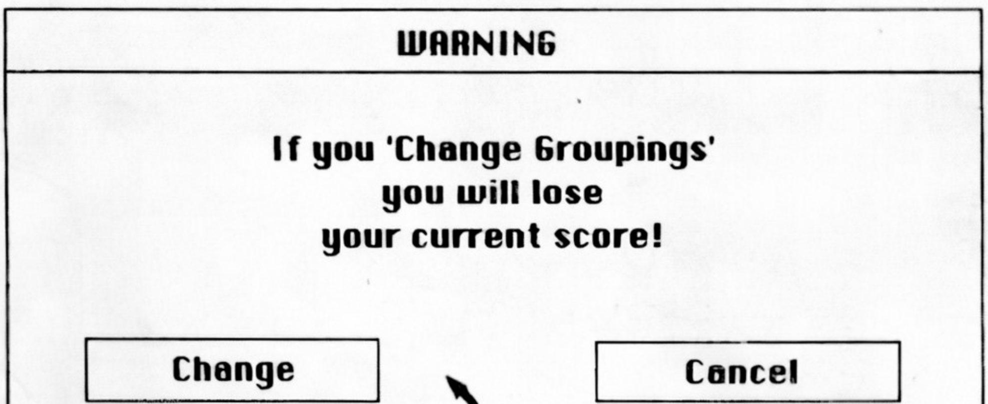
Hangnem
C-dúr
A-moll

Szín
Előtér
Háttér

A következő dolgokat tudjuk itt beállítani:

- A C-64 tizenhat színe közül választhatunk ki egyet-egyet a háttér, ill. az előtér számára. Színmódosításkor a fekete nyilat a táblázat jobb alsó részén látható fehér nyilak közül a megfelelőhöz téve kell megnyomni a RETURN billentyűt.

- A kottasorok kétféle elrendezése közül választhatunk. Ha a táblázat bal oldalán látható dupla kottasor feletti kis négyzetet sötétítjük be, akkor 2x2 kottasor, míg, ha a jobb oldali négyzetet sötétítjük be, akkor 3x1 kottasor kerül majd a képernyőre. Ahhoz azonban, hogy tényleg megváltozzon a sorok elrendezése, még egy dolog szükséges: a fekete nyilat a táblázat bal felső sarkában található kis négyzetre mozgatva meg kell nyomni a RETURN billentyűt. Ha az előzőleg használt kottasor-elrendezéstől eltérőt választunk, akkor a következő figyelmeztetés jelenik meg:



FIGYELMEZTETÉS

Ha megváltoztatod a csoportosítást, elveszíted az aktuális kottát!

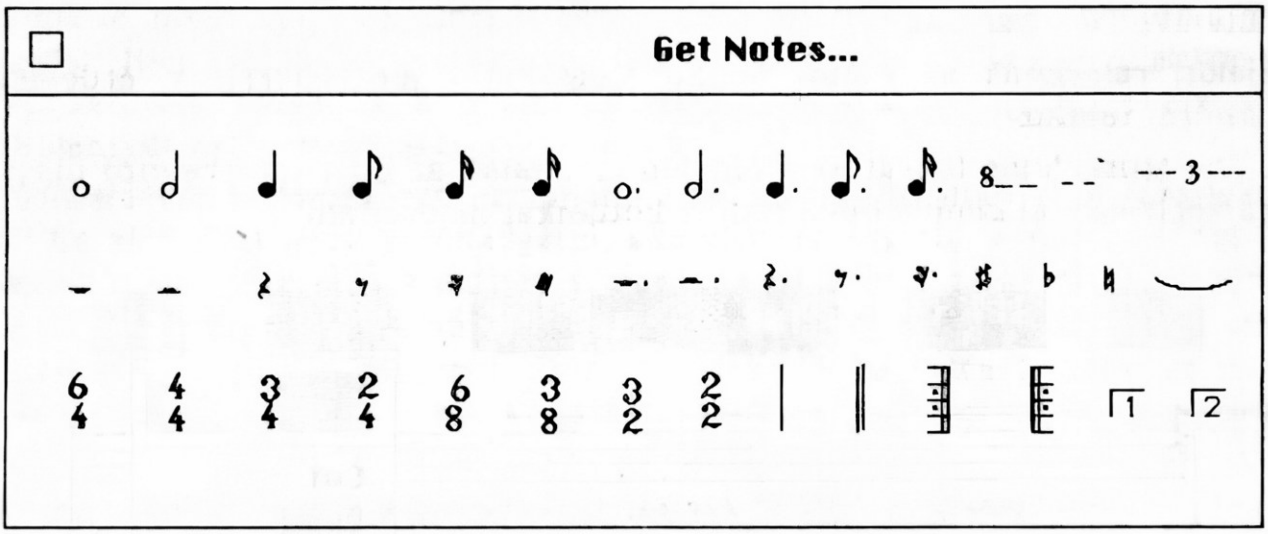
Így kottaíráskor a megfelelő kottasor-elrendezést kell elsőként meghatároznunk.

- A kottában kétféle kulcs szerepelhet: violinkulcs és basszuskulcs (C-kulcsot sajnos nem használhatunk).

A Setup Screen táblázat bal alsó, Key Signature felirattal jelzett részéről még nem esett szó. Innen tudjuk leolvasni, hogy az egyes előjegyzések (legfeljebb 6 db ♯ vagy 6 db ♭) milyen hangnemet jelentenek. Ha például nincs előjegyzés, akkor, - mint már az I. fejezetben volt róla szó - beszélhetünk C-dúr és A-moll hangsorról is.

Get Notes...F1 (Kottajelek)

Mint a cím is jelzi, a következő táblázathoz nemcsak úgy juthatunk el, hogy a Tools szolgáltatásai közül a Get Notes-t választjuk ki, hanem egyszerűbben is: elég csupán az F1-et megnyomnunk.



A táblázatban szereplő jelek közül úgy tudunk választani, hogy a fekete nyilat rámozgatjuk a megfelelőre, majd megnyomjuk a RETURN-t. Ekkor eltűnik a táblázat, s a kiválasztott jelet a fekete nyíllal a kívánt helyre tudjuk vinni.

A jel elhelyezésénél segítségünkre lehet a képernyő bal felső sarkában olvasható útmutatás. Ez a fekete nyíl függőleges irányú mozgatásával állandóan változik, s azt mutatja, hogy éppen melyik kottasornál, s milyen hangnál járunk. Ha pl. **3G** látható a bal felső sarokban, akkor ez azt jelenti, hogy a harmadik kottasorhoz tartozó G hangra mutat a nyíl hegye, azaz a RETURN megnyomásakor ide kerül a kottafej.

A kottákban leggyakrabban előforduló jeleket nemcsak a Get Notes szolgáltatás segítségével, hanem közvetlenül a billentyűzetről is kezelhetjük. A következő táblázat erről nyújt bővebb felvilágosítást:

	1	2	3	4	5	6	7	8	S	F	N	B	U	T
	𝅝	𝅗𝅥	♩	♪	𝅘𝅥𝅯	𝅘𝅥𝅰			♯	♭	♮	|	♩↔♩	♩t
SHIFT	𝄻	𝄼	𝄽	𝄾	𝄿	𝅀								
C-	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]						

Ha kottaíráskor pl. A SHIFT és a 3. billentyűt egyszerre nyomjuk meg, akkor a fekete nyíl által mutatott helyre egy negyed értékű szünetjel kerül. (A táblázat utolsó sorában szereplő jelekről a későbbiekben lesz szó.)

Az S, F, N ill. a B billentyű nem a véletlen szeszélye folytán kapta a táblázatból leolvasható funkcióját:

♯ = **s**harp sign

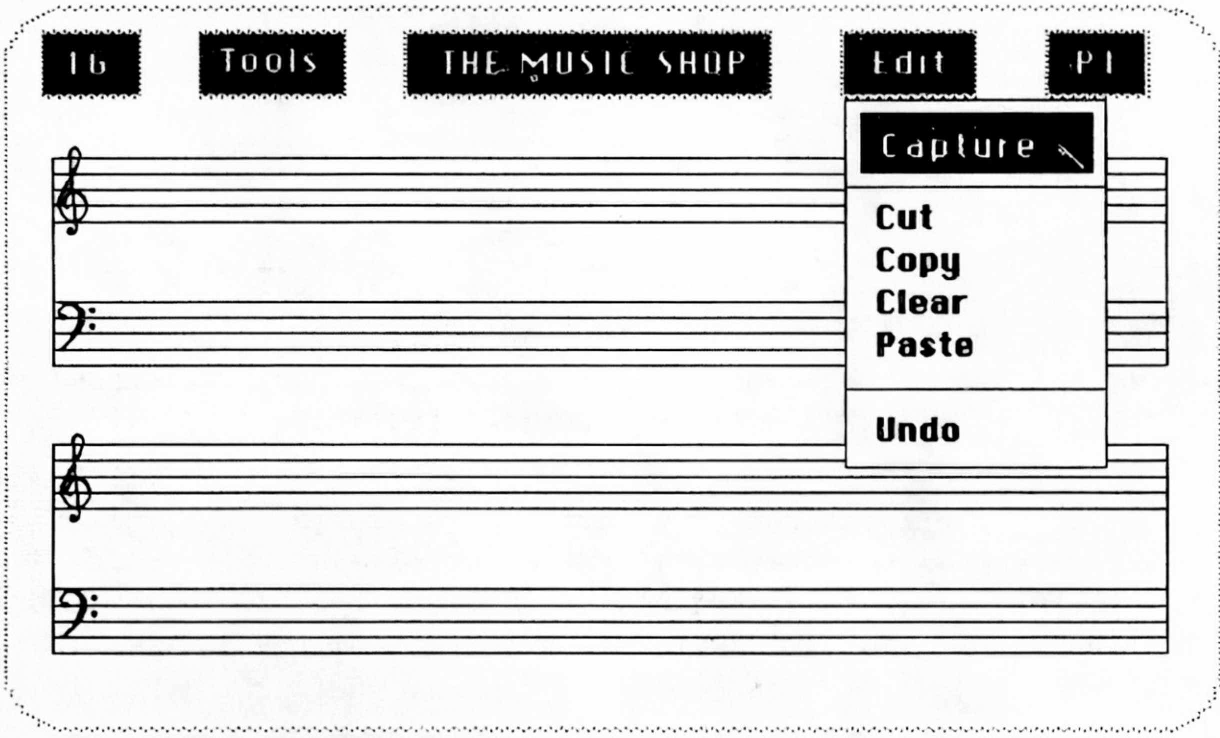
♭ = **f**lat sign

♮ = **n**atural sign

ütemvonal = **b**ar line

Mielőtt rátérnénk az Adjust Sound szolgáltatás megismerésére, érdemes egy rövid kitérőt tennünk.

A The Music Shop felirat jobb oldalán találhatók az **Edit** (szerkesztő) utasítások, melyek segítségével könnyebben tudjuk kottánkat módosítani.



A **Cut**, **Copy**, ill. a **Clear** utasítások segítségével egy kottasor tetszés szerinti részét kivághatjuk, átmásolhatjuk, ill. törölhetjük. Ahhoz, hogy e három utasítás valamelyikét használni tudjuk, először azt kell meghatároznunk, hogy a kotta mely részére vonatkozzon. Ezt a célt szolgálja a **Capture** utasítás, melynek segítségével a kotta bármely részét bekeretezhetjük, "elfoghatjuk".

Copy esetén azt is meg kell adnunk, hogy hova akarjuk átmásolni a kiválasztott kottarészletet. Ezt a **Paste** utasítás segítségével tehetjük meg. A megjelenő kis ecsetet arra a helyre kell mozgatnunk, ahonnan kezdve a választott részt át szeretnénk másolni. A Paste utasítást nemcsak a Copy hanem a Cut és a Clear utasítások után is használhatjuk, azok megismétlésére.

Az **Undo** segítségével a legutolsó műveletet megelőző állapotot hozhatjuk vissza.

Nagy előny, hogy a Capture és a Paste is közvetlenül meghívható a billentyűzetről a C, ill. a P billentyű megnyomásával.

Még néhány fontos tudnivaló, melyek ismerete egyszerűbbé teszi a kottaírást:

1. A már elkészült kottarész bármely jelét magunkkal vihetjük, ha rámutatunk a nyíllal, és egyszer lenyomjuk a RETURN-t.

2. Ha a RETURN-t egymás után kétszer nyomjuk meg, akkor eltűnik az a jel, amelyre a nyíl mutat.

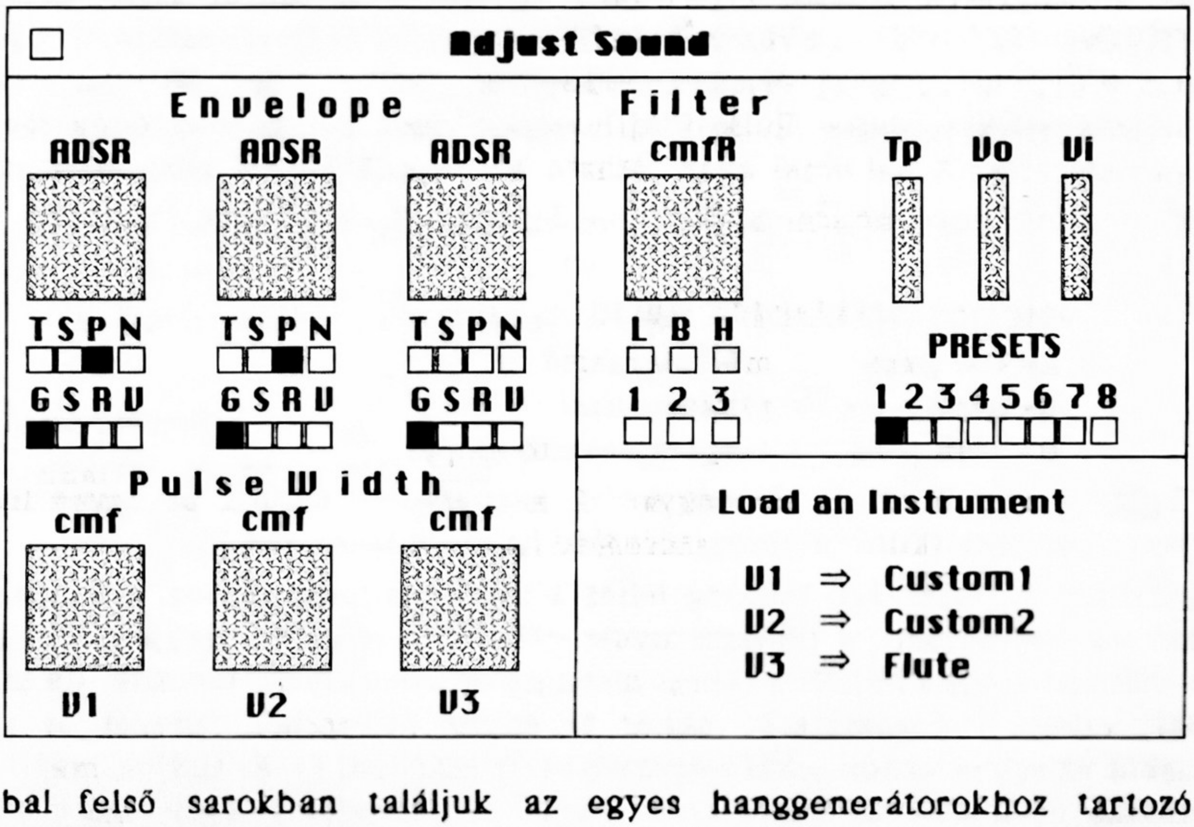
3.Az INST/DEL billentyűt ugyanúgy használhatjuk, mint programozáskor.

4. A kotta számára 20 oldalt biztosít a program. A képernyő jobb felső sarkából olvashatjuk le, hogy hányadik oldal látható. Pl. **P3** azt jelenti, hogy a 3. lapot látjuk (page=oldal). Ha a kottában előre akarunk lapozni, akkor az F3, ha hátra, akkor az F5 billentyűt kell megnyomnunk. Ha azonnal az első lapra akarunk ugrani, akkor a CTRL és az S billentyűt kell egyszerre megnyomnunk.

5. Kottaíráskor érdemes egy-egy újabb rész elkészülte után is meghallgatni a darabot. Ha elölről akarjuk meghallgatni, akkor vagy vigyük a nyilat a bal felső sarokba, majd nyomjuk meg a RETURN-t, vagy pedig a nyíl helyzetétől függetlenül üssük le az F7 billentyűt. Hosszabb darab írásakor sok időt venne igénybe, ha mindig elölről hallgatnánk meg a dallamot. Szerencsére az F8 hatására tetszőleges hangtól kezdve is megszólaltathatjuk. A kotta lejátszásakor a bal felső sarokban egy jel látható.

A kitérő után folytassuk a zenélés eszközeinek (Tools) megismerését!

Adjust Sound (A hangzás beállítása)



A bal felső sarokban találjuk az egyes hanggenerátorokhoz tartozó ADSR értékeket. Mint tudjuk, ezek 0 és 15 között változhatnak. Ez alatt a hullámformát választhatjuk ki az egyes hanggenerátorok számára:

T= triangle	háromszög
S= sawtooth	fűrészfog
P= pulse	négyszög
N= noise	(fehér) zaj

A 2. fejezetben már volt arról szó, hogy a zajt nem szabad más hullámformával keverni. Így, ha a zajt választjuk ki, a többi hullámforma automatikusan kikapcsolódik.

A következő sorban szereplő betűk jelentése:

G= gate-bit	kapu bit
S= synchronization	szinkronizáció
R= ring modulation	körmoduláció
V= vibration	vibráció

Mivel a kapu bittel, a szinkronizációval és a körmodulációval a korábbi fejezetekben már részletesen foglalkoztunk, most csak a vibrációval kell megismerkednünk.

A vibráció lényegében nem más, mint a hang lebegtetése, az alaphang frekvenciájának gyors változtatása egy kisebb intervallumon belül. A vibráció nagyságát 0 és 15 között változtathatjuk a táblázat jobb felső részén látható Vi-vel jelzett skálán. Ennek baloldalán még két "magányos" skála áll.

Tp= tempo	a dallam tempója
Vo= volume	hangerő

A Filter szó jelentését azok is könnyen kitalálhatják, akik nem tudnak angolul, elég csak a füstszűrős, filteres cigarettára gondolni. A felirat alatti skálákon a levágási frekvencia (c,m,f) és a rezonancia (R) nagyságát tudjuk beállítani. A levágási frekvencia a $(16^2 \cdot c+16 \cdot m+f)$ értékkel arányosan változik. Ugyanígy határozhatjuk meg az impulzusszélességet – Pulse Width – is. (Ennek persze csak négyszöghullám esetén van szerepe.) A bal oldal alján látható V_1, V_2, ill. V_3 azt jelzi, hogy az egyes oszlopok paraméterei rendre az első, második, ill. harmadik hanghoz (voice) tartoznak.

Az L, B, H betűk a szűrés fajtáját mutatják:

L= low pass	mélyáteresztő
B= band pass	sáváteresztő
H= high pass	magasáteresztő szűrés

Az 1, 2, 3 számok alatti kis négyzetek segítségével tudjuk az egyes hangokat szűrésre kiválasztani. (Külső jel megszűrésére itt nincs lehetőség.)

Egy dallam lejátszásakor szükség lehet a tempó, a hangerősség, a hangszín, stb. módosítására. Szerencsére a táblázat nyolc változatát is képes megjegyezni a Music Shop. A változatokhoz a PRESETS felirat alatti nyolc kis négyzet tartozik. Ha átállítjuk a táblázat valamely paraméterét, akkor az éppen bekapcsolt "preset" is változik. Kottaíráskor az egyes változatokat négyzetbe tett számmal (1-8) tudjuk meghívni. Ha egy kottában pl. a [2] jellel találkozunk, akkor ez azt jelenti, hogy innen kezdve a második előre elkészített hangszín, tempó, stb. együttesnek megfelelően fog szólni a dallam. Ilyen jelet úgy tudunk a kottába írni, hogy a C= és a megfelelő szám-billentyűt egyszerre nyomjuk meg.

Zeneboltunkban több hangszer is található, erre hívja fel a figyelmet a "Load an Instrument" azaz "Tölts be egy hangszert" felirat. Nézzük csak, milyen hangszerek kérhetők!

Oboe	oboa
Bassoon	fagott
Clarinet	klarinét
Trumpet	trombita
Piano	zongora
Harpsicord	csemballó
Organ	orgona
Xylophone	xilofon
Percussion	ütőhangszerek
Cymbal	cintányér
Bells	csőharangok
Flute	furulya, fuvola
Violine	hegedű

Befejezésül még néhány hasznos tudnivaló az Adjust Sound táblázatról:

1. A táblázat F2-vel hívható.
2. Miután a táblázat megjelent a képernyőn, F7-tel elindíthatjuk a gépben lévő kotta lejátszását. Az F7 ismételt megnyomása megállítja a zenét.
3. Ha az Adjust Sound táblázat látható a képernyőn, akkor a dallam folytonosan ismétlődik.
4. A dallam lejátszása közben is módosíthatjuk a táblázat paramétereit, és a hangszereket.
5. A skálák értékét F3-mal lehet növelni, és F5-tel csökkenteni.

Kedves Olvasó!

Sok sikert a további zenéléshez!

Fourier tétele

Fourier tételének lényege az, hogy a rezgések felbonthatók harmonikus rezgések (szinusz rezgések) összegére. Általában bármilyen f(t) periodikus folyamat, amelynek periódusa T - azaz T az a legkisebb időköz, amelyre teljesül, hogy $f(t+T) = f(t)$ minden t-re - egyértelműen előállítható megfelelő amplitúdókkal rendelkező harmonikus rezgések összegeként. Ezek körfrekvenciái az adott f(t) rezgés körfrekvenciájának ($\omega = 2\pi/T$) egész számú többszörösei:

$$f(t) = c_0 + c_1 \sin(\omega t + \gamma_1) + c_2 \sin(2\omega t + \gamma_2) + \ldots + c_n \sin(n\omega t + \gamma_n) + \ldots$$

azaz $$f(t) = c_0 + \sum_{i=1}^{\infty} c_i \sin(i\omega t + \gamma_i)$$

Az – általában végtelen – Fourier-sorok c_i és γ_i paraméterei matematikailag meghatározhatók.

1a. A háromszöghullám Fourier sorának előállítása

Az előbbi képlet a következő alakban is felírható:

$$f(t) = a_0/2 + a_1 \cos \omega t + a_2 \cos 2\omega t + \ldots + a_n \cos n\omega t + \ldots + b_1 \sin \omega t + b_2 \sin 2\omega t + \ldots + b_n \sin n\omega t + \ldots$$

ahol $a_0 = 2c_0$, $a_i = c_i \sin \gamma_i$ és $b_i = c_i \cos \gamma_i$ $i = 1, 2, \ldots$

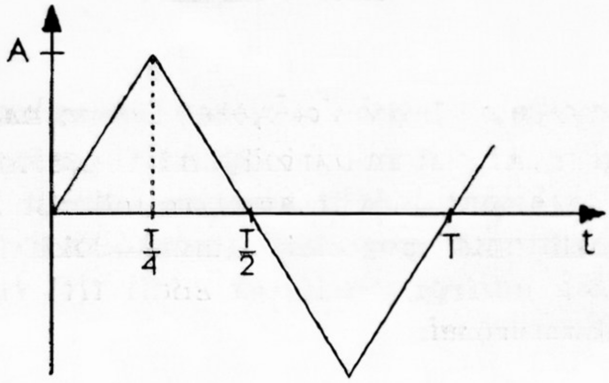
(Itt egyszerűen csak a $\sin(\alpha+\beta) = \sin\alpha \cos\beta + \cos\alpha \sin\beta$ azonosságot használtuk fel.

Az f(t) előállításában szereplő a_i, b_i együtthatókat a következő módon határozhatjuk meg:

$$a_i = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cos i\omega t \, dt$$

$$b_i = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \sin i\omega t \, dt$$ ahol T az f(t) periódusa

Mivel a háromszöghullám páratlan függvény, azaz $f(x) = -f(-x)$, ezért a Fourier-sorban koszinuszos tagok nem fognak szerepelni.



A szimmetrián alapuló megfontolások miatt elegendő csak egy negyedperiódust megvizsgálni. A háromszöghullám a [0, T/4] tartományban a következő egyszerű alakba írható:

$$f(t) = \frac{t}{\frac{T}{4}} \cdot A = \frac{4At}{T} \qquad 0 \le t \le \frac{T}{4}$$

Így az i-edik harmonikus amplitúdója:

$$b_i = \frac{8}{T} \int_0^{\frac{T}{4}} f(t) \sin i\omega t \, dt = \frac{8 \cdot 4 \cdot A}{T^2} \int_0^{\frac{T}{4}} t \sin i\omega t \, dt$$

De $T = \frac{2\pi}{\omega}$, ezért

$$b_i = \frac{8 \cdot 4 \cdot A \cdot \omega^2}{4\pi^2} \int_0^{\frac{\pi}{2\omega}} t \cdot \sin i\omega t \, dt$$

Az integrál kiszámítását parciális integrálással oldjuk meg:

$$b_i = \frac{8A}{\pi^2}\,\omega^2\left(\left[\frac{-t\cos i\omega t}{i\omega}\right]_0^{\frac{\pi}{2\omega}} - \int_0^{\frac{\pi}{2\omega}} \frac{-\cos i\omega t}{i\omega}\,dt\right) =$$

$$= \frac{8A}{\pi^2}\,\omega^2\left(-\frac{\pi\cos i\frac{\pi}{2}}{2\,i\omega^2} - 0 - \left[\frac{-\sin i\omega t}{i^2\omega^2}\right]_0^{\frac{\pi}{2\omega}}\right)$$

De $\cos i\frac{\pi}{2} = 0$, így

$$b_i = \frac{8A}{\pi^2}\,\omega^2\left(\frac{\sin i\frac{\pi}{2}}{i^2\omega^2} - \frac{\sin 0}{i^2\omega^2}\right)$$

De $\sin 0 = 0$, ezért

$$b_i = \frac{8A}{\pi^2 i^2}\sin i\frac{\pi}{2} \qquad i=1,2,\ldots$$

$$\text{Azonban } \sin i\frac{\pi}{2} = \begin{cases} +1 & \text{ha } i\equiv 1 \bmod 4 \\ 0 & \text{ha } i\equiv 2 \bmod 4 \text{ vagy } i\equiv 0 \bmod 4 \\ -1 & \text{ha } i\equiv 3 \bmod 4 \end{cases}$$

Így pedig valóban csak a páratlan tagok maradnak, s a háromszöghullám Fourier-sora:

$$X(t) = \frac{8A}{\pi^2}\left(\sin\omega t - \frac{1}{3^2}\sin 3\omega t + \frac{1}{5^2}\sin 5\omega t - \ldots\right)$$

azaz

$$X(t) = \frac{8A}{\pi^2}\cdot\sum_{i=0}^{\infty}\frac{(-1)^i\sin(2i+1)\,\omega t}{(2i+1)^2}$$

Ebből a sorbafejtésből látszik, hogy a háromszöghullám alaphangjához csak azok a felhangok csatlakoznak, melyeknek frekvenciája az alaphang frekvenciájának páratlan számú többszöröse, sőt az is, hogy a felhangok amplitúdói a frekvencia növekedésével négyzetesen csökkennek.

1b. Egy feladat a fűrészfoghullám Fourier-sorával kapcsolatban. (Az alábbiak azokhoz szólnak, akik szeretik a matematikát, de még nem tanultak integrálásról, így nem tudták követni az előbbi gondolatmenetet.)

A fűrészfoghullám Fourier-sora a következő:

$$X(t)=\frac{2\cdot A}{\pi}(\sin\omega t-\quad\sin 2\omega t+\quad\sin 3\omega t-\dots)$$

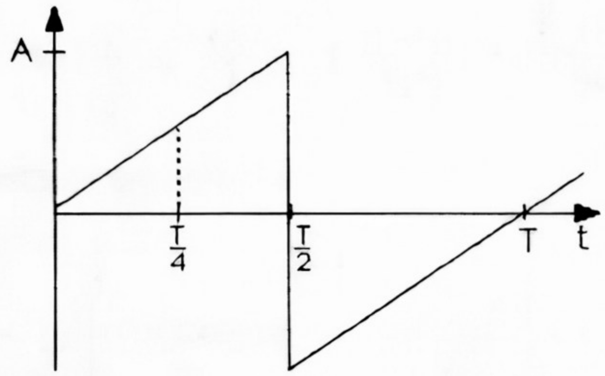
azaz

$$X(t)=\frac{2\cdot A}{\pi}\cdot\sum_{i=1}^{\infty}\frac{(-1)^{i+1}\sin i\omega t}{i}$$

azaz

$$X(t)=k\cdot(\sin\omega t-\frac{1}{2}\sin 2\omega t+\frac{1}{3}\sin 3\omega t-\dots)$$

Hogyan lehetne igazolni, hogy k valóban $2A/\pi$-vel egyenlő? Mindenki gondolkozzon el a problémán, és csak utána olvassa el a most következő megoldást.



Az a kellemetlen a dologban, hogy $f(T/2)\neq A$, hanem $f(T/2)=0$. Ezért az első ötlet az, hogy a $T/4$ helyen vizsgáljuk a függvény értékét, a második pedig az, hogy legyen $T=2\pi$. Ez utóbbit minden további nélkül feltételezhetjük. Ekkor:

$$f(\frac{T}{4})=f(\frac{\pi}{2})=k\cdot(\sin\frac{\pi}{2}-\frac{1}{2}\sin\frac{2\pi}{2}+\frac{1}{3}\sin\frac{3\pi}{2}-\dots)=$$

$$=k\cdot(\sin\frac{\pi}{2}+\frac{1}{3}\sin\frac{3\pi}{2}+\frac{1}{5}\sin\frac{5\pi}{2}+\dots)=$$

$$=k\cdot(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\dots)$$

Ez viszont éppen $k\cdot\pi/4$. Ennek bizonyításától most eltekintünk. Az érdeklődők a [9] 186-187. oldalán utánanézhetnek a bizonyításnak. Érdemes megjegyezni, hogy az $1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\dots=\frac{\pi}{4}$ összefüggést szokás Leibniz formulának is nevezni.

Ezek szerint:

$$f(\frac{T}{4})=k\cdot\frac{\pi}{4}\Rightarrow A=2\cdot f(\frac{T}{4})=2\cdot\frac{k\pi}{4}=\frac{k\pi}{2}\Rightarrow k=\frac{2A}{\pi}$$

Ezzel valóban sikerült igazolni állításunkat.

A háromszöghullámmal ellentétben, a fűrészfoghullám nemcsak a páratlan felhangokat tartalmazza, hanem a párosakat is, az amplitúdók lineárisan csökkennek a frekvencia növekedésével.

1c. Fourier-sorok grafikus ábrázolása

A következő program segítségével, melyet SIMON's BASIC-ben írtam, ki tudjuk rajzolni a képernyőre a szinusz-görbét, a szabályos négyszög-, háromszög-, valamint a fűrészfoghullámot. Ehhez persze ismernünk kell a négyszöghullám Fourier-sorát is.

A szabályos, 50%-os négyszöghullám Fourier-sora a következő:

$$X(t)=\frac{4A}{\pi}\left(\sin\omega t+\frac{1}{3}\sin 3\omega t+\frac{1}{5}\sin 5\omega t+\ldots\right)$$

azaz

$$X(t)=\frac{4A}{\pi}\cdot\sum_{i=0}^{\infty}\frac{\sin(2i+1)\omega t}{2i+1}$$

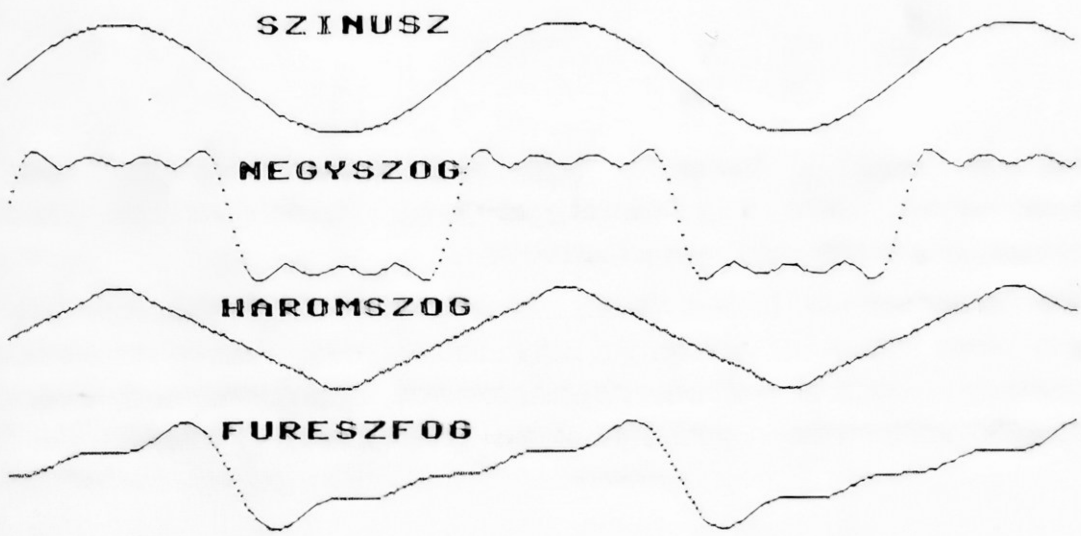
Tehát most is csak a páratlan felharmonikusok szólalnak meg, mint háromszöghullám esetén, azzal a különbséggel, hogy most ezek amplitúdói csak lineárisan csökkennek a frekvencia növekedésével.

A következő programban a négyszög, háromszög és fűrészfoghullámnak csak három felhangját, azaz a Fourier-soroknak csak az első négy tagját tanulmányozzuk (32, 42, ill. 52. sorok). Érdemes e sorok változtatásával megfigyelni azt, hogy hogyan változik az egyes hullámformák képe a felhangok számának növelésével.

```
10 COLOUR 0,0:HIRES 3,0
20 TEXT 70,2,"SZINUSZ",1,1,8
21 FORX=1TO300
22 Y=25-SIN(X/20)*20
23 PLOT X,Y,1
24 NEXT
25 Y=0
30 TEXT 65,55,"NEGYSZOG",1,1,8
31 FORX=1TO300
32 FORI=1TO7 STEP2
33 Y=Y+(1/I)*SIN(I*X/20)
34 NEXTI
35 Y=75-Y*4*20/π
36 PLOTX,Y,1
37 Y=0
38 NEXTX
39 TEXT 60,105,"HAROMSZOG",1,1,8
40 K=1
41 FORX=1TO300
42 FORI=1TO7 STEP2
43 Y=Y+K*(1/I↑2)*SIN(I*X/20)
44 K=-K:NEXTI
45 Y=120-Y*8*20/π↑2
46 PLOTX,Y,1
47 Y=0:K=1
48 NEXTX
49 TEXT 60,150,"FURESZFOG",1,1,8
```

```
50 K=1
51 FORX=1TO300
52 FORI=1TO4
53 Y=Y+K*(1/I)*SIN(I*X/20)
54 K=-K:NEXTI
55 Y=170-Y*2*20/π
56 PLOTX,Y,1
57 Y=0:K=1
58 NEXTX
60 COPY
70 GOTO70
```

A programban $\omega t = x/20$ és az amplitúdó A=20. A 60-as sorban szereplő COPY utasítás segítségével a nyomtatóra is ki tudjuk rajzolni a különböző grafikonokat. Ennek eredményét láthatjuk a következő ábrán:

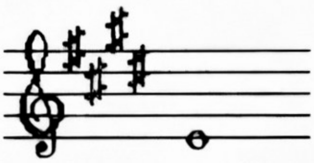


Programunk jó példa arra, hogy milyen nagy hasznát lehet venni a számítógépnek különböző függvények, grafikonok ábrázolása esetén. A matematika órán se feledkezzünk meg erről.
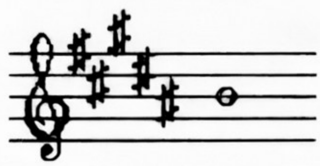
Megoldások

Feladat 1.

a/



E = dó



H = dó

b/ Amikor egy újabb bé előjegyzést teszünk ki, akkor ezzel a pillanatnyi ti hangot kell tá-ra módosítanunk, hogy továbbra is megmaradjon a dúr hangsor.

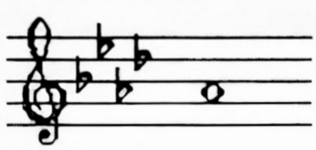
d 2 **r** 2 **m** 1 **f** 2 **s** 2 **l** 2 **t** 1 **d**

d 2 **r** 2 **m** 1 **f** 2 **s** 2 **l** 1 **ta** 2 **d**
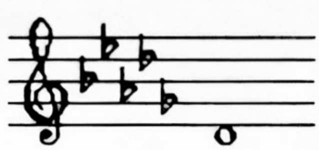
A módosított hangsorban az előbbi fá veszi át a dó szerepét. Így az is igaz, hogy amit most módosítottunk, az lesz egy újabb bé esetén a dó helye.

Ha gyorsan meg akarnánk mondani, hogy pl. a 6 bé esetén hol a dó, akkor a következő módon lehet okoskodni:

ha nincs előjegyzés, akkor a C a dó. Minden újabb bé esetén 5 félhanggal kerül magasabbra a dó helye, így a 6. bé után a C-nél 6*5=30 félhanggal lesz magasabb a dó, viszont a C-nél 2*12=24 félhanggal magasabb hang szintén C, tehát csak azt kell megmondanunk, hogy melyik az a hang, amelyik 30-24=6 félhanggal nagyobb a C-nél. Ez pedig a <u>Gesz</u> (mivel a b lefelé módosít). Tehát 6 bé előjegyzés esetén Gesz a dó. Hasonlóan, ha például 7 bé az előjegyzés, akkor 7*5-2*12=35-24=11, következésképp <u>Cesz</u> a dó.



Asz = dó



Desz = dó

Feladat 2.

A 23-24. oldalon szereplő frekvenciatáblázatot a következő programmal készítettük:

```
1 OPEN4,4:CMD4
3 PRINT"-------------------------------------------------------------------"
5 PRINT"   HANG    (HZ)   PARA-    F.B  A.B      P2      F2    A2"
6 PRINTTAB(18)"METER"
7 PRINT"-------------------------------------------------------------------"
20 DIM HB(95,1),HC(95,1),NEV$(11)
22 FORI=0TO11:READNE$(I):NEXT
```

```
23 FORI=0TO8:READNS$(I):NEXT
24 FR=15.4338531:REM 440/2↑(58/12)
25 FORI=0TO95
26 FR=FR*2↑(1/12)
27 FK=INT(FR*10+0.5)/10
30 P1=278.460278*2↑(I/12)+0.5:REM 7493/2↑(57/12)
35 HB(I,1)=INT(P1/256)
40 HB(I,0)=INT(P1-256*HB(I,1))
45 P2=268.237281*2↑(I/12):REM 36376/2↑(85/12)
50 HC(I,1)=INT(P2/256)
55 HC(I,0)=INT(P2-256*HC(I,1))
60 IFI>=10THEN 62
61 PRINT" "I;:GOTO 63
62 PRINT I;
63 M=INT(I/12):L=I-12*M
64 N$=NE$(L)+NS$(M)
65 PRINTN$;
66 IF LEN(N$)=3THENPRINT" ";
68 IF FK>999.9THEN PRINTFK;:GOTO72
69 IF FK>99.9THEN PRINT" "FK;:GOTO72
70 PRINT"  "FK;
72 IF FK=INT(FK)THENPRINT"  ";
73 P1=INT(P1):P2=INT(P2)
74 IF P1>9999THEN PRINT" " P1;:GOTO80
75 IF P1>999THEN PRINT"  "P1;:GOTO80
76 PRINT"   "P1;
80 IF HB(I,1)>99THEN PRINT" "HB(I,1);:GOTO90
82 IF HB(I,1)>9 THEN PRINT"  "HB(I,1);:GOTO90
84 PRINT"   "HB(I,1);
90 IF HB(I,0)>99THEN PRINTHB(I,0);:GOTO100
92 IF HB(I,0)>9 THEN PRINT" "HB(I,0);:GOTO100
94 PRINT"  "HB(I,0);
100 IF P2>9999THENPRINT" "P2;:GOTO110
102 IF P2>999THENPRINT"  "P2;:GOTO110
104 PRINT"   "P2;
110 IF HC(I,1)>99THEN PRINT" "HC(I,1);:GOTO120
112 IF HC(I,1)>9 THEN PRINT"  "HC(I,1);:GOTO120
114 PRINT"   "HC(I,1);
120 IF HC(I,0)>99THEN PRINTHC(I,0):GOTO130
122 IF HC(I,0)>9 THEN PRINT" "HC(I,0):GOTO130
124 PRINT"  "HC(I,0)
130 NEXT
135 PRINT# 4
140 CLOSE4
300 DATA "C","C#","D","D#","E","F"
301 DATA "F#","G","G#","A","A#","H"
302 DATA "-0","-1","-2","-3","-4","-5"
303 DATA "-6","-7","-8"
400 END
```

Feladat 3.

A program futása alatt nem szól hang, aminek az az oka, hogy a GATE-bit gyors nyitása-zárása esetén nem kezdődik el még a felfutás (A) sem. A program megállításakor, ha a 35-ös vagy a 40-es soron áll meg a program, akkor a SID az első három fázist hajtja végre, s a hang nem is akar elhalkulni, hiszen a kapu zárva van (GATE-bit=1). Ha a 45-ös soron áll meg a program, akkor rögtön a negyedik szakasz hajtódik végre.

Feladat 5.

A hang azért nem szólal meg – annak ellenére, hogy minden szükséges paramétert beállítottunk –, mert az AD, SR paraméterek csak a kontroll-regiszter (hullámforma+GATE-bit magas) beállítása után kaptak értéket. Így a helyes megoldáshoz akkor jutunk, ha a 25-ös sort átsorszámozzuk 30-40 közöttire (=35), és ezután a 25-ös sort töröljük. Tanulság: a paraméterek beállításának sorrendje is számít.

```
10 S=54272
15 FORI=0TO23:POKES+I,0:NEXT
20 POKES+24,15
30 POKES+5,255:POKES+6,143
32 POKES+4,17
35 POKES+1,19:POKES,121
40 FORI=1TO10000:NEXTI
45 POKES+4,32
50 END
```

Feladat 7.

Egy 11 jegyű bináris szám maximális értéke $2^{11}-1 = 2047$. Így, ha a beállított 11 jegyű számunk értéke X, akkor a hozzá tartozó frekvencia kb.:

$$X\cdot\frac{12000-30}{2047}+30 \approx 5{,}85\cdot X+30$$

Fordítva, ha kívánt frekvencia Y, akkor a beállítandó paraméterek értéke kb.:

$$\frac{y-30}{5{,}85} \qquad 30 \le y \le 12000 \text{ (Hz)}$$

Így, ha Y=440 Hz, akkor $410/5{,}85 \approx 70$-nek kell lennie a 11 jegyű számunk értékének, következésképpen:

F = INT (70/8) = 8 kerül a SID 22. regiszterébe

és A = 70-8×F = 6 kerül a SID 21. regiszterébe.

Feladat 8.

Ez a program arra példa, hogy hogyan változik a szűrés levágási frekvenciájának növekedésével a megszólaló hang hangspektruma, hangszíne.

10 S= a SID 0. regiszterének címe

15 a SID írható regisztereinek nullára állítása

20 31=15+16 => maximális hangerő és aluláteresztő szűrés

25 A=0, D=9, S=10, R=12 a II. hang esetén

30 a II. hang impulzusszélessége 50%-os, azaz szabályos

35 II. hang szűrőn át

40 J ötösével nő 255-ig

45 a levágási frekvencia kb. 5,85 x (8xJ) + 30

50 a II. hang hullámformája négyszöghullám és a GATE-bit magas

55 a II hang magassága a 220 Hz frekvenciájú A-3

60 a hang kitartása

65 a "kaput" kinyitjuk (GATE-bit=0), s elkezdődik a hang elengedése (4. fázis)

70 ciklus vége

Feladat 9.

A különböző lehetőségek közül kettőt mutatunk be:

1.: POKE S+23, PEEK(S+23)- INT(PEEK(S+23)/16)x16+10x16

2.: POKE S+23,(PEEK(S+23) OR (10x16)) AND (10x16+15)

Feladat 12.

```
10 REM METRONOM
15 PRINT"♥"
20 S=54272:TI$="000000"
25 FORI=1TO24:POKES+I,0:NEXT
30 POKES+24,15
35 POKES+5,11
40 INPUT"1 PERC ALATT HANYSZOR";H
45 K=3600/H
50 T=TI
55 POKES+1,20:POKES+4,33
60 IFTI<T+K THEN 60
65 POKES+4,32:GOTO50
```

Ehhez a programhoz mindössze annyit kell tudni, hogy a C-64 belső órája, a TI másodpercenként kb 60-szor, vagyis percenként 3600-szor növeli eggyel az értékét. Így, ha azt akarjuk, hogy percenként H-szor jelezzen a metronómunk, akkor a jelzéseknek akkor kell bekövetkezniük, amikor a TI értéke K=3600/H-val nagyobb, mint az ezt megelőző jelzésnél levő TI érték.

Feladat 13.

A 213-as, s a 218-as sorban szereplő 10-es értéknek 11-nek kellene lennie a kotta szerint. Érdemes a 211, 216 sorokban szereplő 11-est is 10-re módosítva meghallgatni a programunkat.

Feladat 14.

Felesleges a HB tömböt 1-95-ig feltölteni. A programban előforduló legmélyebb hang HB szerinti sorszáma 36, a legmagasabbé pedig 7+51+12=70, így a 30-as sort FOR I = 36 TO 70-re módosítva időt takarítunk meg.

Feladat 15.

Az 1113-1114-es sorokat csak akkor hajtja végre a progam, ha az ISZ(H) értéke nulla. Ez viszont azt jelenti, hogy az impulzusszélesség alsó byte-jának értékét mutató szám még 255, s ezt kell 0-ra módosítanunk. Azért van szükség a 1113-1114 sorokra, hogy a nulla kinyomtatása után ne 055 jelenjen meg a képernyőn. A két ötös törlését egy print utasítással elvégezhetjük. Töröljük az 1114-es sort, és az 1113-as sorban szereplő PRINT " " helyett írjunk PRINT "   "-et.

Még egy dolgot észrevehetünk. Mivel a változók értékének nyomtatásakor egy hely mindig kimarad az előjel számára, így az 1116-os sorban az A=10×H hatására nem a 10×H-nál, hanem a 10×H+1-nél kezdődik a kiírás, hiszen pozitív számokat nyomtatunk.

Feladat 17.

Egyszerű a magyarázat. A HB tömb ugyanis 1-től kezdve van feltöltve (33-as sor), míg a NEV$ tömb 0-tól.

Feladat 21.

Igen lényeges a sorok szerepe, ugyanis ezek biztosítják azt, hogy ne álljon meg a gép hibaüzenettel, ha olyan billentyűről, oktávról indítottuk el a beépített dallamot, melynek hangjai túl magasak lennének a C-64 számára.

Ilyen biztonsági részeket érdemes az első két beépített dallamhoz is írni.

Irodalomjegyzék

[1] DR.ÜRY L.: Commodore 64 BASIC felhasználói kézikönyv (LSI ATSZ Bp. 1984.)

[2] Commodore 64 User's Guide (Commodore Business Machines, Inc. 1984.)

[3] Commodore 64 Bedienungshandbuch (Commodore Electronics Limited, 1984.)

[4] VOGEL J. - CRIMSCHAW: Commodore 64 Musich Buch (Birkhauser, 1984.)

[5] G.LUDINSKI: Brainteasers for the Commodore 64 (Phoenix Publishing Associates)

[6] DR.BUDÓ Á.: Kísérleti Fizika I. kötet (Tankönyvkiadó, Bp. 1975.)

[7] DARVAS G.: A zene anatómiája (Zeneműkiadó, Bp. 1973.)

[8] KRUZA R. - BANAY G.: A szintetizátor a zenei gyakorlatban (Zeneműkiadó, Bp. 1985)

[9] Elemi matematika IV.
/Szerkesztette: Hortobágyi István/ (Tankönyvkiadó, Bp. 1978)

[10] Ezer év kórusa /Forrai M. gyűjteménye/ (Editio Musica Bp. 1977)

[11] A mi dalaink /Összeállította: Ugrin G./ (Tankönyvkiadó, Bp. 1979)

[12] MusiCalc (Synthetizer & Sequencer 1 – Owner's Manual)

[13] M.L. DE JONG.: Assembly language programming with the C-64
(Brady Communications Company, Inc. 1984)

[14] M.ENGLAND - D.LAWRENCE: Machine Code Graphics and Sound on the C-64

ZENE KEDVELŐKNEK
C-64/128

FIGYELEM !

A Commodore 128 számítógép C-64-es üzemmódjában az itt közölt valamennyi program működik.

A Commodore 128 számítógép C-128-as üzemmódjában (BASIC V7.0) az összes olyan BASIC program működik, amelyik csak a SID chip regisztereit használja. Ebben az esetben a dallamokat a gép lassabban játssza le, mert az interpreter lassabb mint a C 64-é. A FAST: RUN parancsok kiadása után viszont gyorsabbak lesznek a dallamok, mert 2 MHz-es órajellel üzemel a gép. A zene ritmusát megadó értékek módosításával korrigálni lehet a gép sebességéből adódó eltéréseket.

LSI

LSI ALKALMAZÁSTECHNIKAI
TANÁCSADÓ SZOLGÁLAT

Digital Drums

Wildboys (1985)

Ez a program különböző ritmus-alapok előállítását teszi lehetővé. A dobolás hangzását a programba beépített analizált dob-hangok memóriában tárolt hullámformáinak az egymás utáni kiolvasásával és a hanggenerátorba táplálásával digitálisan állítja elő.

A programnak három fő üzemmódja van van:

- billentyűzet üzemmód
- joystick üzemmód
- programozási mód

Billentyűzet üzemmód:
Belépés: SYS 30208

A billentyűzet üzemmódból nem használható ki a program minden lehetősége, de a hangszínek teljes skáláját elérhetővé teszi. A programozási üzemmódtól eltérően, a hanghosszúság nincs külön meghatározva – de erre nincs is szükség, hiszen a billentyűzeten a hanghosszúságokat és a ritmust ízlés szerint játszhatjuk.

A különféle dob-hangokat a következő billentyűk szólaltatják meg:

2	3	4	5	6	7	8	9	0	+	-
Q	W	E	R	T	Y	U	I	O	P	@
basszus-dob	Hi hat (hosszú)	Hi hat (rövid)	snare (pergődob)	Tama 1	Tama 2	Tama 3	Tama 4	kopogódob	kolomp	cintányér

Az egymás alá eső billentyűk egyazon dob-hangot eredményeznek; a gombok megduplázása a dobok pergetésében segít, mert egy billentyűt nem tudnánk elég sűrűn nyomkodni. Mutató és középső ujjunkat az egymás feletti billentyűkre helyezve viszont könnyen játszhatunk gyors és egyenletes pergetést. Próbáljuk ki!

Joystick üzemmód:
Belépés: SYS 29184

A joystick üzemmód használatakor két joystickre van szükség, de igazán még így sem lehet jól játszani ebben az üzemmódban. Elképzelhető, hogy esetleg más, a joystick-bemenetekre csatlakozó perifériák számára írták a programba. A joystick meghúzására nagyon gyorsan pereg a dob, több irány esetén viszont kellemetlen berregő hang keletkezik.

Programozási üzemmód:

A belépés annyiból áll, hogy aktivizálni kell a BASIC-kiegészítést. Ehhez a SYS 29696 utasítást kell kiadnunk, ezután a BASIC-et a szokásos módon

használhatjuk, de kibővített utasításkészlettel. Ezért a program három üzemmódja közül ez nyújtja a legtöbb felhasználási lehetőséget.

Fontos megkötés, hogy az első programsornak egy REM utasításnak kell lennie!

A dob-utasítások mind $ jellel kezdődnek, amit egy betű követ: ez utal a megszólaltatni kívánt dob-fajtára. Ezután egy szám jön, ami az adott hangszernek a különböző hangszínű, hangmagasságú, lecsengési idejű változatai közül kijelöli az általunk kívántat. Néhány példa:

$B1	basszus dob
$B2	mélyebb hangú basszus dob
$H1	Hi hat cintányér (hosszú)
$H2	Hi hat cintányér (rövid)
$S1	Snare dob (hosszú)
$S2	Snare dob (rövid)
$T1-$T4	különböző magasságú Tama dobok
$K1	kopogó hangú dob
$K2	kolomp
$C1	cintányér (hosszú)
$C1	cintányér (rövid)
$P	egy egységnyi szünet

Sajnos, a dobok zengési ideje nem arányosan változik, ez nehezíti a pontos ritmusok létrehozását.

A lejátszás sebességét a 752-es cím átírásával (2–9) változtathatjuk.

Bár a program nem írja, a tapasztalatom az, hogy a 752-es és 753-as címek egyidejű nullázásával érhetünk el egyenletes, könnyen szabályozható dob-hosszúságokat.

Végül két példa:

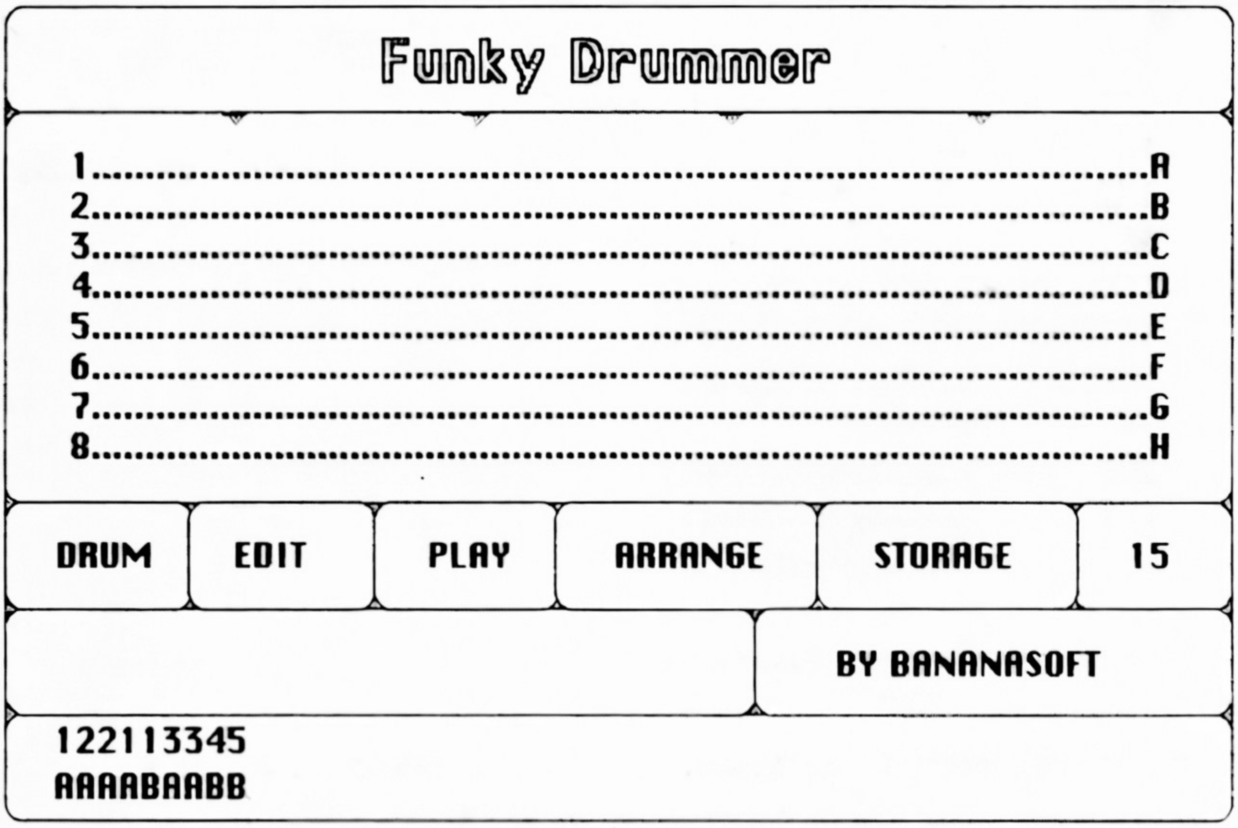
```
20 SYS 29696:POKE 753,0:POKE 752,0
30 $B1:FORI=1 TO 100:$P:NEXT:$H2
40 FOR I=1 TO 100:$P:NEXT:$B1:FORI=1 TO 100:$P:NEXT:$H2
50 FOR I=1 TO 100:$P:NEXT:$B1:FORI=1 TO 100:$P:NEXT:$H2
60 FOR I=1 TO 100:$P:NEXT:$B1:FORI=1 TO 100:$P:NEXT:$H2
65 FOR I=1 TO 100:$P:NEXT
70 FOR I=1 TO 3:$T1:NEXT
80 FOR I=1 TO 3:$T2:NEXT
90 FOR I=1 TO 3:$T3:NEXT
100 FOR I=1 TO 3:$T4:NEXT
110 FOR I=1 TO 13:$S2:NEXT:$C1:$C1:$C1
```

```
20 SYS 29696:POKE 752,0:POKE 753,0
25 FOR A=1 TO 3
30 $T1:FOR I=1 TO 10:$P:NEXT
40 $T1:FOR I=1 TO 10:$P:NEXT
50 $T4:FOR I=1 TO 10:$P:NEXT
60 $T4:FOR I=1 TO 10:$P:NEXT
65 NEXT
70 $S2:FOR I=1 TO 20:$P:NEXT
80 $K1:FOR I=1 TO 10:$P:NEXT
90 $K1:FOR I=1 TO 40:$P:NEXT
100 FOR I=1 TO 50 $P:NEXT
110 GOTO 25
```

Funky Drummer

Bananasoft (1985)

A Funky Drummer a Digital Drums-hoz hasonlóan dob-hangok megszólaltatására alkalmas. Előnye azzal szemben, hogy időbeosztásban programozható, így nagyon pontos ritmusok állíthatók vele elő. Hátránya viszont, hogy kevesebb dob-típust képes megszólaltatni, cintányér például egyáltalán nincs benne. A program a következő táblázatot mutatja a képernyőn:



A dobszintetizátor 6 funkcióval rendelkezik, melyek mindegyikét a RETURN gombbal indíthatjuk, s az S billentyűvel léphetünk ki belőle. E funkciók:

DRUM: a billentyűzeten játszhatunk (1-8 gombok)

EDIT: saját ritmust állíthatunk össze. A szerkesztőben 8 oldal van, mindegyik oldalon 8 sorral, egy sor 4 ütem. Itt is 8 féle dob-hang áll a rendelkezésünkre.

A kurzort a CRSR billentyűvel mozgathatjuk, a kívánt dob megszólalását pedig a hozzá tartozó szám-billentyű lenyomásával írhatjuk be a táblára. A ⇐ gombbal törölhetünk egy hangot, az oldalakat az A-H betűkkel válthatjuk, s a P billentyű lenyomásával játszhatunk le egy beírt sort. Egy egész oldalt a SHIFT és a CLR gombok együttes lenyomásával lehet törölni.

PLAY: az EDIT és az ARRANGE funkciókban összeállított dob-partitúra megszólaltatása

ARRANGE: itt határozhatjuk meg, hogy az egyes sorokat milyen sorrendben akarjuk megszólaltatni. Egy sort az azt tartalmazó oldal betűjelével, s a kívánt sor számával jelölhetünk ki. A képernyőn a vízszintes kurzor-gombbal mozoghatunk.

törölni pedig a SPACE-el tudunk. Ha egy beírt minta végére *-ot írunk, akkor a lejátszáskor a csillaghoz érve a sor elejére ugrik vissza a program, és végtelen ciklusban játssza majd azt a szakaszt.

STORAGE: a háttértároló(k) használatát teszi lehetővé. Először egy LOAD OR SAVE? kérdés jelenik meg a képernyőn, amire L-el vagy S-el kell válaszolnunk, attól függően, hogy betölteni vagy kimenteni szeretnénk egy ritmust. Majd a DISK OR TAPE? kérdésre kell D-vel vagy T-vel válaszolnunk, annak megfelelően, hogy lemezegységgel vagy magnóval dolgozunk. Ezután az üzenet-mező törlődik, és beírhatjuk a betölteni vagy kimenteni kívánt program nevét. A kért művelet a RETURN gomb lenyomásával veszi kezdetét.

A **sebesség-mező**be írt számmal állíthatjuk be a a lejátszás gyorsaságát: 00, a leggyorsabb, és 99, a leglassúbb lejátszásnak megfelelő számok közti tetszőleges értéket írhatunk be. A CRSR◊ billentyűvel egyesével felfelé, a SHIFT+CRSR◊ billentyűvel egyesével lefelé állíthatjuk a sebesség értékét.

Ára: 125 ,- Ft

Központi Statisztikai Hivatal
Számítástechnikai és
Ügyvitelszervező Vállalat



BÍZZA A SZÜV-RE

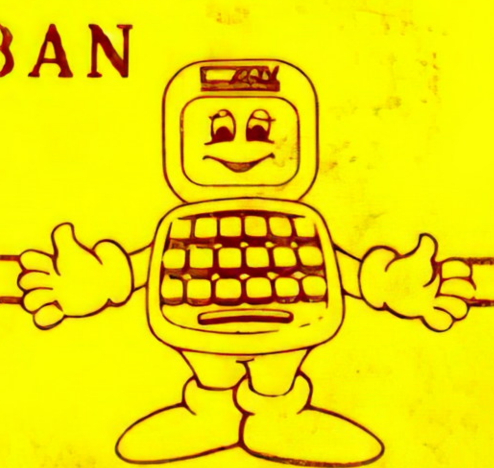
szakmai problémáit,
hardver és szoftver igényeit !

MINDENKIHEZ KÖZEL VAGYUNK

ügyfélszolgálati irodáinkban
ingyenes tanácsadás
helyszíni gépbérlet
adathordozók és segédanyagok,
szakirodalom árusítása
oktatás, betanítás

egyéni és közületi felhasználók részére
az ország egész területén

KOMPLEX SZOLGÁLTATÁS FŐVÁLLALKOZÁSBAN



COMPUTER-M